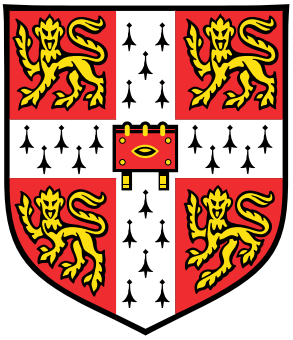
Ensemble generation and compression for speech recognition



Jeremy Heng Meng Wong

Department of Engineering

University of Cambridge

This dissertation is submitted for the degree of

Doctor of Philosophy

St. Edmund's College September 2018

Gloriam Deo, et in Terra pax.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables, and equations, and has fewer than 150 figures.

Jeremy Heng Meng Wong
September 2018

Acknowledgements

Thank you to my family and friends for your continued support through this PhD. Thank you to my supervisor, Mark Gales, for teaching me how to do research. Thank you in alphabetical order to Jeff Chen, Geraldine Duggan, Rachel Fogg, Patrick Gosling, Peter Grandi, Adnan Haider, Diane Hazell, Kate Knill, Anna Langley, Yanmin Qian, Anton Ragni, Louise Segar, Matt Shannon, Haipeng Wang, Yu Wang, Phil Woodland, Justin Yang, and Chao Zhang for the working infrastructure, exploration of ideas, and teaching. This PhD has been funded by the National Science Scholarship, from the Agency for Science, Technology, and Research in Singapore.

Summary

For many tasks in machine learning, performance gains can often be obtained by combining together an ensemble of multiple systems. In Automatic Speech Recognition (ASR), a range of approaches can be used to combine an ensemble when performing recognition. However, many of these have computational costs that scale linearly with the ensemble size. One method to address this is teacher-student learning, which compresses the ensemble into a single student. The student is trained to emulate the combined ensemble, and only the student needs to be used when performing recognition. This thesis investigates both methods for ensemble generation and methods for ensemble compression.

The first contribution of this thesis is to explore approaches of generating multiple systems for an ensemble. The combined ensemble performance depends on both the accuracy of the individual members of the ensemble, as well as the diversity between their behaviours. The structured nature of speech allows for many ways that systems can be made different from each other. The experiments suggest that significant combination gains can be obtained by combining systems with different acoustic models, sets of state clusters, and sets of sub-word units. When performing recognition, these ensembles can be combined at the hypothesis and frame levels. However, these combination methods can be computationally expensive, as data is processed by multiple systems.

This thesis also considers approaches to compress an ensemble, and reduce the computational cost when performing recognition. Teacher-student learning is one such method. In standard teacher-student learning, information about the per-frame state cluster posteriors is propagated from the teacher ensemble to the student, to train the student to emulate the ensemble. However, this has two limitations. First, it requires that the teachers and student all use the same set of state clusters. This limits the allowed forms of diversities that the ensemble can have. Second, ASR is a sequence modelling task, and the frame-level posteriors that are propagated may not effectively convey all information about the sequence-level behaviours of the teachers. This thesis addresses both of these limitations.

The second contribution of this thesis is to address the first limitation, and allow for different sets of state clusters between systems. The proposed method maps the state cluster posteriors from the teachers' sets of state clusters to that of the student. The map is derived by

considering a distance measure between posteriors of unclustered logical context-dependent states, instead of the usual state cluster. The experiments suggest that this proposed method can allow a student to effectively learn from an ensemble that has a diversity of state cluster sets. However, the experiments also suggest that the student may need to have a large set of state clusters to effectively emulate this ensemble. This thesis proposes to use a student with a multi-task topology, with an output layer for each of the different sets of state clusters. This can capture the phonetic resolution of having multiple sets of state clusters, while having fewer parameters than a student with a single large output layer.

The third contribution of this thesis is to address the second limitation of standard teacher-student learning, that only frame-level information is propagated to emulate the ensemble behaviour for the sequence modelling ASR task. This thesis proposes to generalise teacher-student learning to the sequence level, and propagate sequence posterior information. The proposed methods can also allow for many forms of ensemble diversities. The experiments suggest that by using these sequence-level methods, a student can learn to emulate the ensemble better. Recently, the lattice-free method has been proposed to train a system directly toward a sequence discriminative criterion. Ensembles of these systems can exhibit highly diverse behaviours, because the systems are not biased toward any cross-entropy forced alignments. It is difficult to apply standard frame-level teacher-student learning with these lattice-free systems, as they are often not designed to produce state cluster posteriors. Sequence-level teacher-student learning operates directly on the sequence posteriors, and can therefore be used directly with these lattice-free systems.

The proposals in this thesis are assessed on four ASR tasks. These are the augmented multi-party interaction meeting transcription, IARPA Babel Tok Pisin conversational telephone speech, English broadcast news, and multi-genre broadcast tasks. These datasets provide a variety of quantities of training data, recording environments, and speaking styles.

Keywords: Teacher-student, ensemble, automatic speech recognition, random forest, sequence discriminative training

Table of contents

List of figures

List of tables

Acronyms

AMI Augmented Multi-party Interaction.

ASR Automatic Speech Recognition.

BLSTM Bi-directional Long Short-Term Memory.

CD Context-Dependent.

CE Cross-Entropy.

CMLLR Constrained Maximum Likelihood Linear Regression.

CN Confusion Network.

CNC Confusion Network Combination.

CTC Connectionist Temporal Classification.

DCT Discrete Cosine Transform.

DFT Discrete Fourier Transform.

DNN Deep Neural Network.

EM Expectation Maximisation.

ESN Echo State Network.

FER Frame Error Rate.

G2P Grapheme-to-Phoneme.

GMM Gaussian Mixture Model.

HMC Hybrid/Hamiltonian Monte Carlo.

HMM Hidden Markov Model.

IHM Individual Headset Microphone.

KL Kullback-Leibler.

LF Lattice-Free.

LM Language Model.

LSTM Long Short-Term Memory.

MAP Maximum a-Posteriori.

MBR Minimum Bayes' Risk.

MFCC Mel-Frequency Cepstral Coefficients.

MGB Multi-Genre Broadcast.

ML Maximum Likelihood.

MMI Maximum Mutual Information.

MSE Mean Squared Error.

MT Multi-Task.

NN Neural Network.

PDT Phonetic Decision Tree.

PLP Perceptual Linear Predictive.

RBM Restricted Boltzmann Machine.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

ROVER Recogniser Output Voting Error Reduction.

SAT Speaker Adaptive Training.

SER Sentence Error Rate.

SGD Stochastic Gradient Descent.

SGLD Stochastic Gradient Langevin Dynamics.

sMBR state-level Minimum Bayes' Risk.

TCD Tree Cluster Divergence.

TDNN Time-Delay Neural Network.

TS Teacher-Student.

VLLP Very Limited Language Pack.

WER Word Error Rate.

WFSA Weighted Finite State Acceptor.

WFST Weighted Finite State Transducer.

Notation

General Notations

s	a scalar is represented by a lower-case, non-bold, and italic letter
$\boldsymbol{\omega}$	a vector or sequence is represented by a lower-case and bold letter
$\mathbf{W}$	a matrix is represented by an upper-case and bold letter
$\boldsymbol{\Phi}$	a vector of model parameters is represented by an upper-case and bold Greek letter
$\odot$	element-wise multiplication
$\boldsymbol{\nabla}_{\boldsymbol{\Phi}}\mathcal{F}$	derivative of $\mathcal{F}$ with respect to vector $\boldsymbol{\Phi}$
$s \in \mathcal{T}$	set of state clusters defined by the leaves of the decision tree
$\mathcal{G}_{\omega}$	set that can represent $\boldsymbol{\omega}$

Variables

a	lattice arc
$\mathbf{b}$	neural network bias vector
c	logical context-dependent state
h	neural network hidden layer activation
$\mathbf{I}$	identity matrix
o	observation feature
s	state cluster
$\mathbf{W}$	neural network weight matrix

z	neural network hidden layer pre-nonlinearity activation
Z	normalisation factor
α	forward probability
β	backward probability
γ	language scaling factor
η	learning rate
$\boldsymbol{\Theta}$	student's model parameters
κ	acoustic scaling factor
λ	ensemble interpolation weight
$\boldsymbol{\mu}$	Gaussian mean
ν	discount factor
$\boldsymbol{\Xi}$	classifier model parameters
π	Dropout rate
$\boldsymbol{\Sigma}$	Gaussian covariance
χ	criteria interpolation weight
$\boldsymbol{\Psi}$	feature extractor model parameters
ω	word
$\mathcal{C}$	set of logical context-dependent states
$\mathcal{D}$	training data
$\mathcal{M}$	model topology and other system design choices
$\mathbb{A}$	lattice
$\mathbb{H}, \mathbb{C}, \mathbb{L}, \mathbb{G}$	weighted finite state transducer or acceptor graphs

Functions

g	neural network activation function

δ	Kronecker delta-function
φ	feature extractor
$\mathcal{A}$	acoustic score
$\mathcal{F}$	training criterion
$\mathcal{L}$	risk function for minimum Bayes' risk
$\mathcal{O}$	computational complexity
$\mathcal{R}$	regularisation term in training criterion
$\mathcal{T}$	phonetic decision tree
$\mathbb{D}$	general distance measure
$\mathbb{E}$	expectation

Probabilities

P	discrete probability distribution
p	probability density function
q/Q	approximate probability density function/distribution
$\mathcal{N}$	Gaussian probability density function
$\mathcal{U}$	uniform distribution or probability density function

indexes

k	neural network layer
K	number of neural network hidden layers
l	word sequence position index
L	length of utterance
m	ensemble member index
M	ensemble size
r	utterance index

R total number of utterances

t frame index

T total number of frames

v training iteration

Superscripts and Subscripts

$\mathbf{s}_{1:T}$ sequence with length T

s_t tth element in vector or sequence

$\mathbf{\Phi}^m$ mth member of an ensemble

$\mathbf{\Phi}^{(v)}$ vth iteration

$\mathbf{h}^{(k)}$ kth layer in a neural network

$\boldsymbol{\omega}^{\text{ref}}$ reference targets

$\boldsymbol{\omega}^*$ 1-best

$s^{\mathbf{\Theta}}$ state cluster using the student's decision tree

Accents

$\widehat{\mathbf{\Phi}}$ set of all model parameters in the ensemble, $\widehat{\mathbf{\Phi}} = \left\{\mathbf{\Phi}^1, \cdots, \mathbf{\Phi}^M\right\}$

$\widetilde{p}$ pseudo likelihood

$\widetilde{\mathcal{D}}$ mini-batch of data

ó features that have been projected through a neural network feature extractor

$\overline{\mathbf{\Phi}}$ weighted average of model parameters

Chapter 1

Introduction

Human society is characterised, not by the actions of isolated individuals, but by the complex interactions and relationships between people. Verbal communication through speech and language forms a natural backbone, upon which these interactions and relationships are established and maintained. With the ease at which verbal communication can be used to convey complex ideas, it seems a logical extension to use speech, not only for communication between people, but also as an interface between humans and machines. Any verbal human-computer interface first requires an automated process to map from a spoken audio waveform to text, before this text can be processed by a downstream application. The process of performing this mapping is Automatic Speech Recognition (ASR).

1.1 Automatic speech recognition

ASR technology has seen much progress, since the early days of recognising isolated digits in clean speech, by matching spectrograms to a limited set of speech patterns [31]. Modern methods tackle the harder problem of recognising continuous speech over a large vocabulary, in more varied environments. This is often approached by treating the task of ASR as a statistical problem and using probabilistic models. The Hidden Markov Model (HMM)-based approach [3, 118] uses separate probabilistic models to capture the acoustics, temporal alignment, and language aspects of speech. Progress in recent years has seen a shift from Gaussian Mixture Model (GMM) acoustic models [76] and n-gram language models [2], to deep learning methods, using multi-layer Neural Networks (NN) [8, 12]. The greater parameter sharing in NN models allows the parameters to be estimated more reliably from a limited quantity of data. The NN acoustic model parameters can be learned from data by using the frame-level cross-entropy criterion [12]. Sequence discriminative training methods [1, 77] can also be used, and these have often been shown to outperform frame-level training

[86]. Rather than training separate acoustic, alignment, and language models, methods have also been proposed to train a single model that maps directly from the input acoustic observation sequence to a sequence of characters or words [20, 54].

1.2 Ensemble methods and teacher-student learning

With these and other improvements to ASR technology, the state-of-the-art ASR performance is gradually approaching those of human annotators [161]. The disagreement between the transcriptions by different expert human annotators [95] can be viewed as the gold standard for ASR performance. A technique that is often used to obtain these state-of-the-art performances is to combine together an ensemble of multiple systems [161]. The combined performance of an ensemble depends on both the individual system performances and the diversity between the system behaviours [61]. Previous work has investigated making the systems behave differently by using different acoustic models [33], sets of state clusters [135], feature representations [126], and sets of sub-word units [150]. This thesis investigates these and other methods for generating a diverse ensemble.

However, an ensemble can be computationally expensive to use to perform recognition, as data is required to be fed through multiple systems and multiple decoding runs are needed. Thus, although an ensemble can be used to obtain good ASR performance, the computational cost can hinder its deployment in practical ASR applications, particularly in situations where ASR is required to be run on devices with limited hardware resources. As such, there is practical interest in reducing the computational cost of performing recognition, by compressing the ensemble.

Teacher-student learning [17] is one possible method that can be used for ensemble compression. Here, a single student is trained to emulate the combined behaviour of the ensemble. Only this single student needs to be used for recognition, instead of the ensemble of multiple systems. The standard teacher-student learning method [93] trains the student by propagating per-frame state cluster posterior information from the ensemble. This requires that all systems in the ensemble must use the same set of state clusters, and therefore limits the allowed forms of diversities that the ensemble can have. Furthermore, these frame-level posteriors may not adequately convey information about the sequential nature of speech data.

This thesis generalises the teacher-student learning framework to allow it to be used to compress ensembles with more forms of diversities. The teacher-student learning method is also generalised to the sequence level, by taking inspiration from previous work that has shown that sequence discriminative training can often outperform frame-level cross-entropy training [86]. The proposed methods allow information about the sequence-level

behaviours of diverse teachers to be propagated, and may allow a more diverse ensemble to be more effectively compressed into a single student. The sequence-level teacher-student learning method may even be used to propagate information between systems with completely different architectures, such as between HMM-based systems and end-to-end NN systems.

Although current ASR systems can perform well on clean speech data [161], it is still a challenge to recognise speech in noisy environments and from a distance. Domain adaptation methods aim to train systems to be more robust, when used across a variety of different recording environments. In addition to ensemble compression, teacher-student learning can also be used for domain adaptation [75, 92]. Here, a student is trained to emulate the behaviour of a teacher, where the student and teacher take as inputs synchronised parallel data from different domains, such as from noisy and clean recording environments respectively. Although it is not investigated here, the methods proposed to generalise teacher-student learning in this thesis can also be applied when using teacher-student learning for domain adaptation.

1.3 Contributions

The combined performance of an ensemble of multiple systems depends on both the individual system performances, as well as the diversity between the system behaviours. When generating an ensemble for ASR, there are many ways to make the systems behave differently. The first contribution of this thesis is to analyse the diversities and combination gains that can be obtained for a range of possible ensemble generation methods.

An ensemble can be computationally expensive to use for recognition, and teacher-student learning is one method that can be used to reduce this cost. However, the standard teacher-student learning method requires that all systems within the ensemble use the same set of state clusters, thereby restricting the allowed forms of ensemble diversity. The second contribution of this thesis generalises frame-level teacher-student learning to allow for a diversity of state cluster sets within the teacher ensemble. This work is published by the author of this thesis in [155, 156].

The standard teacher-student learning method also only trains the student using a frame-level criterion. However, the per-frame posterior information that is propagated may not effectively capture the sequence-level behaviours of the teachers, which may be important for the sequence modelling task of ASR. The third contribution of this thesis generalises the teacher-student learning framework to use sequence-level criteria, and propagate sequence posterior information from the teachers to the student. These sequence-level criteria also have the potential to allow for more forms of diversities within the teacher ensemble than

those allowed by frame-level teacher-student learning. This work is published by the author of this thesis in [154].

1.4 Thesis organisation

This thesis is organised as follows. Chapter 2 presents an overview of HMM-based ASR. Performance gains can often be obtained by combining an ensemble of multiple systems. Chapter 3 reviews possible methods of generating such a diverse ensemble. However, an ensemble can be computationally expensive to use to perform recognition. Chapter 4 describes possible approaches to compress an ensemble and reducing this cost. One possible compression method is teacher-student learning. However, the standard teacher-student learning method limits the allowed forms of ensemble diversities and only propagates frame-level information. Chapter 5 proposes extensions to frame-level teacher-student learning, to allow the ensemble to have a diversity of state cluster sets, and also to allow lattice-free systems to be used. Chapter 6 proposes other forms of information that can be propagated, namely hidden layer representation information and sequence-level information, the latter of which generalises teacher-student learning to the sequence level. Chapter 7 describes the tasks and setups used for the experiments. The experiments in Chapter 8 investigate the different forms of diversities and combination methods that an ensemble can use. Chapter 9 assess the ability of a frame-level student to emulate a teacher ensemble, and also the proposed extension to allow for different sets of state clusters between the student and teachers. The experiments in Chapter 10 investigate propagating hidden layer and sequence-level information from the teachers to the student. Chapter 11 implements the proposed methods within a lattice-free framework. Finally, Chapter 12 summaries the work and suggests possible directions for future research.

Chapter 2

Models for speech recognition

Automatic Speech Recognition (ASR) is a sequence-to-sequence modelling task. A probabilistic framework can be used, such that an ASR system maps from a sequence of input acoustic observations,

$$\mathbf{O}_{1:T} = \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T, \tag{2.1}$$

to a hypothesised word sequence at the output

$$\boldsymbol{\omega}_{1:L} = \omega_1, \omega_2, \cdots, \omega_L, \tag{2.2}$$

using the hypothesis posterior distribution, $P(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T})$. Here, T is the number of time frames and L is the number of words in an utterance.

This chapter presents an overview of the Hidden Markov Model (HMM)-based approach to ASR. Section 2.1 discusses how recognition can be performed when given hypothesis posteriors. Section 2.2 describes how hypothesis posteriors can be computed in an HMM-based system, which consists of language, alignment, and acoustic models. Sections 2.3, 2.4, and 2.5 describe how the acoustic models can be trained. As opposed to the generative modelling of speech in an HMM-based system, Section 2.6 discusses several discriminative models that can also be used for ASR.

2.1 Recognition

ASR can be described as a discriminative task, where the system classifies each acoustic observation sequence into possible word sequence hypotheses. Given a hypothesis posterior, $P(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T})$, one possible method of performing recognition is to choose the most probable,

or Maximum a-Posteriori (MAP), hypothesis,

$$\boldsymbol{\omega}^*_{\text{MAP}} = \arg\max_{\boldsymbol{\omega}} P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}\right). \tag{2.3}$$

According to Bayes' decision rule [10], choosing the MAP hypothesis minimises the expected classification error rate of the word sequence hypothesis. This can be seen explicitly by re-writing the MAP decoding criterion in the form of

$$\boldsymbol{\omega}^*_{\text{MAP}} = \arg\min_{\boldsymbol{\omega}'} \sum_{\boldsymbol{\omega}} \left[1 - \delta\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right)\right] P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}\right), \tag{2.4}$$

where

$$\delta\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right) = \begin{cases} 1 & \text{, if } \boldsymbol{\omega} = \boldsymbol{\omega}' \\ 0 & \text{, otherwise} \end{cases} \tag{2.5}$$

is the Kronecker δ-function. This is related to the Sentence Error Rate (SER), which measures the classification error rate between hypotheses, $\boldsymbol{\omega}^*_r$, and references $\boldsymbol{\omega}^{\text{ref}}_r$, as

$$\text{SER} = \frac{1}{R}\sum_{r=1}^{R}\left[1 - \delta\left(\boldsymbol{\omega}^*_r, \boldsymbol{\omega}^{\text{ref}}_r\right)\right], \tag{2.6}$$

where r is the utterance index and R is the number of utterances. MAP decoding can be performed using the Viterbi algorithm [146].

The performance of an ASR system is often assessed using the Word Error Rate (WER), rather than the SER. The WER is computed by first aligning the hypothesis to the reference. Then, using a dynamic programming algorithm to compute the word-level Levenstein minimum edit distance,

$$\mathcal{L}_{\text{word}}\left(\boldsymbol{\omega}_{1:L}, \boldsymbol{\omega}'_{1:L'}\right) = N_{\text{Sub}} + N_{\text{Del}} + N_{\text{Ins}}, \tag{2.7}$$

where N_{sub}, N_{del}, and N_{ins} are the numbers of substitution, deletion, and insertion errors respectively, between the aligned word sequences of $\boldsymbol{\omega}_{1:L}$ and $\boldsymbol{\omega}'_{1:L'}$. The WER is the Levenstein distance averaged over all utterances, and normalised by the length of the references,

$$\text{WER} = \frac{1}{\sum_{r'=1}^{R} L^{\text{ref}}_{r'}} \sum_{r=1}^{R} \mathcal{L}_{\text{word}}\left(\boldsymbol{\omega}^*_{r,1:L^*_r}, \boldsymbol{\omega}^{\text{ref}}_{r,1:L^{\text{ref}}_r}\right). \tag{2.8}$$

The WER is often measured against a reference that is provided by manual transcription of the audio. It should be noted that there are often differences between the transcriptions produced by different annotators, even when the annotators are themselves linguistic experts

[95]. The WER between the transcriptions of different expert annotators is often viewed as the gold standard of the performance that can be achieved by an ASR system.

Although the hypothesis that minimises the expected SER can often have a reasonable WER, it may be better to find the hypothesis that instead minimises the expected WER. The hypotheses that minimise these expected error rates can be found by expressing (2.4) more generally as the Minimum Bayes' Risk (MBR) decoding criterion [138],

$$\boldsymbol{\omega}^*_{\text{MBR}} = \arg\min_{\boldsymbol{\omega}'} \sum_{\boldsymbol{\omega}} \mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right) P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}\right). \tag{2.9}$$

It is possible to use a variety of different forms of risk functions, $\mathcal{L}$. Using the word-level Levenstein distance in (2.7) as the risk function minimises the expected WER. As is shown in (2.4), the MAP decoding criterion can be viewed as a special case of the MBR decoding criterion, by using a risk function of

$$\mathcal{L}_{\text{MAP}}\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right) = 1 - \delta\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right). \tag{2.10}$$

MBR decoding can be performed using forward-backward operations over a lattice of competing hypotheses, as is described in [163]. The use of lattices is discussed in Section 2.5.1. It is also possible to obtain the MBR hypothesis by simplifying the lattice into a series of confusion sets, using the Confusion Network (CN) decoding framework [37, 100]. In CN decoding, consecutive words are assumed to be conditionally independent, and the hypothesis posteriors are factorised into a product of word posteriors,

$$P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T}\right) \approx \prod_{l=1}^{L} P\left(\omega_l|\mathbf{O}_{1:T}\right). \tag{2.11}$$

The words in the hypotheses are allocated into the L confusion sets using a heuristic alignment procedure. However, the form of the hypothesis posteriors in (2.11) is limited in its ability to capture the dependencies between consecutive words, provided by the language model. The simplicity of the CN method is that the MBR hypothesis can be obtained by choosing the most probable word within each confusion set,

$$\boldsymbol{\omega}^*_{\text{CN}} = \arg\max_{\omega_1} P\left(\omega_1|\mathbf{O}_{1:T}\right), \cdots, \arg\max_{\omega_L} P\left(\omega_L|\mathbf{O}_{1:T}\right). \tag{2.12}$$

2.2 Modelling structured data

In order to perform recognition, an ASR system is required to produce hypothesis posterior probabilities, $P(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T})$. Speech data naturally has a structured hierarchy, in that a sentence is composed of words, which can be decomposed into sub-word units, which in turn may also be thought of as be being composed of more basic acoustic states. Each of these levels in the hierarchy has its own structure. This structure can be utilised in the design of the ASR system. The Hidden Markov Model (HMM)-based system architecture [3, 118] is one possible method to model these hypothesis posteriors that takes into account the structured nature of speech, by composing together separate models at the different acoustic levels. This section discusses the HMM framework, in a top-down approach along the acoustic hierarchy.

In the HMM framework, the joint probability between hypotheses and observation sequences, $P(\boldsymbol{\omega}_{1:L}, \mathbf{O}_{1:T})$, is captured. This probability density function is generative in nature, as word and observation sequence pairs can be sampled from it. Hypothesis posteriors can be obtained from this joint distribution, as

$$P(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T}) = \frac{P(\boldsymbol{\omega}_{1:L}, \mathbf{O}_{1:T})}{\sum\limits_{\boldsymbol{\omega}'_{1:L'}} P(\boldsymbol{\omega}'_{1:L'}, \mathbf{O}_{1:T})}. \tag{2.13}$$

Performing recognition using either MAP decoding of (2.3) or MBR decoding of (2.9) relies on the accuracy of the hypothesis posteriors. The hypothesis posteriors in (2.13) will be correct, if $P(\boldsymbol{\omega}_{1:L}, \mathbf{O}_{1:T})$ is correct. However, approximations are often made in the modelling of $P(\boldsymbol{\omega}_{1:L}, \mathbf{O}_{1:T})$, which may limit the accuracy of the resulting hypothesis posteriors. It is possible to train an HMM-based system in a discriminative fashion, and several such method are discussed in Section 2.3.3. Section 2.6 considers several discriminative models, that aim to directly model the hypothesis posteriors.

2.2.1 Language model

From the definition of conditional probability, $P(\boldsymbol{\omega}_{1:L}, \mathbf{O}_{1:T})$ can be expressed as

$$P(\boldsymbol{\omega}_{1:L}, \mathbf{O}_{1:T}) = P(\boldsymbol{\omega}_{1:L})\, p(\mathbf{O}_{1:T}|\boldsymbol{\omega}_{1:L}). \tag{2.14}$$

Here, $P(\boldsymbol{\omega}_{1:L})$ is the language model, which computes the prior probability that a sequence of words, $\boldsymbol{\omega}_{1:L} = \omega_l, \cdots, \omega_L$, can occur. This distribution can be factorised into[1]

$$P(\boldsymbol{\omega}_{1:L}) = P(\omega_1) \prod_{l=2}^{L} P(\omega_l | \boldsymbol{\omega}_{1:l-1}) . \tag{2.15}$$

A possible form of language model is the n-gram model [2], which makes the approximation that the current word is conditionally independent of all other words, when given the past $n-1$ words,

$$P(\omega_l | \boldsymbol{\omega}_{1:l-1}) \approx P(\omega_l | \boldsymbol{\omega}_{l-n+1:l-1}, \boldsymbol{\Phi}) . \tag{2.16}$$

The parameters of the n-gram model, $\boldsymbol{\Phi} = \{\pi_{\boldsymbol{\omega}_{1:n}} \quad \forall \boldsymbol{\omega}_{1:n}\}$, separately represent each of the n-gram probabilities,

$$P(\omega_n | \boldsymbol{\omega}_{1:n-1}, \boldsymbol{\Phi}) = \pi_{\boldsymbol{\omega}_{1:n}} . \tag{2.17}$$

These parameters each represent separate entries in a table of probabilities, where the columns and rows are the current word and past $n-1$ words respectively. The language model is generally trained separately from the models in $p(\mathbf{O}_{1:T} | \boldsymbol{\omega}_{1:L})$. The n-gram Language Model (LM) parameters can be trained by maximising the log-likelihood of the manual transcriptions in the training data,

$$\mathcal{F}_{\text{LM-ML}}(\boldsymbol{\Phi}) = \log P\left(\boldsymbol{\omega}_{1:L}^{\text{ref}} \middle| \boldsymbol{\Phi}\right), \tag{2.18}$$

where $\boldsymbol{\omega}_{1:L}^{\text{ref}}$ are the training data manual transcriptions. An implied sum over utterances in the training data is omitted here for brevity. Using this criterion, the n-gram language model parameters can be trained by setting them to

$$\pi_{\boldsymbol{\omega}_{1:n}} = \frac{N_{\boldsymbol{\omega}_{1:n-1}, \omega_n}}{\sum_{\omega_n' \in \mathcal{W}} N_{\boldsymbol{\omega}_{l:n-1}, \omega_n'}}, \tag{2.19}$$

where $N_{\boldsymbol{\omega}_{1:n-1}, \omega_n}$ is the number of times the word sequence $\boldsymbol{\omega}_{1:n-1}$ and word ω_n occur consecutively in the training data, and $\mathcal{W}$ is the vocabulary of all words. However, this maximum likelihood estimate has zero probability for all word sequences that do not appear in the training data, and word sequences with only a few occurrences may not have a reliable probability estimate. Discounting [82] and back-off [88] methods can be used to improve the ability of the n-gram language model to generalise to these unseen or rare word sequences. However, n-gram language model trains a separate parameter for each word sequence. This may not make efficient use of the training data, and as such, a large quantity of training

[1] Often, the transcriptions and hypotheses are padded with additional sentence start and end markers. These markers are not shown in the simplifies representation here.

data may still be required to allow the n-gram language model to generalise well, even with discounting and back-off. When training a language model, data is only required in the form of text, and not audio and text pairs. It is generally cheaper to obtain text data, as manual transcription of audio is not required. As such, a language model can be trained on externally-sourced text data, in addition to the manual transcriptions from the ASR training data.

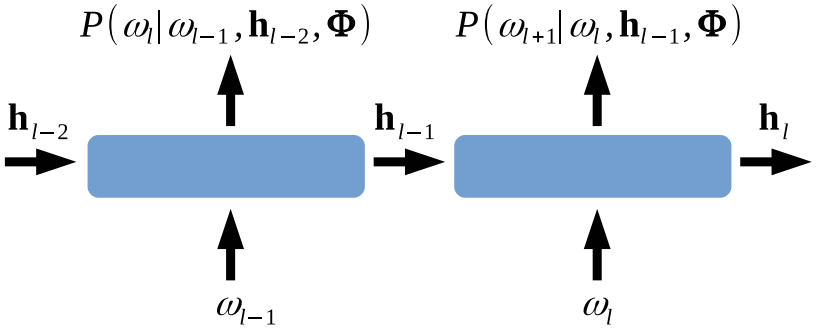


Fig. 2.1 Single layer recurrent neural network language model. Information about the past history of words is stored in the hidden state, $\mathbf{h}_l$.

The n-gram language model suffers from having only a limited context window and a limited ability to generalise to unseen word sequences. In this model, the n-gram probability of every unique sequence of n words is separately estimated. It can be shown that the ability of a language model to generalise can be improved by using a Neural Network (NN) as the language model [8]. The NN parameters are shared across all word sequences, and may therefore be better able to generalise to unseen word sequences. Feed-forward NN language models can take into account a finite context of words, similarly to an n-gram language model. A Recurrent Neural Network (RNN) language model, illustrated in Figure 2.1, can potentially take into account the full history of words [102]. In an RNN language model, the past word history, $\boldsymbol{\omega}_{1:l-2}$, is represented by a hidden state, $\mathbf{h}_{l-2}$, such that the probabilities used in the decomposition of (2.15) are given by

$$P(\omega_l|\boldsymbol{\omega}_{1:l-1}) \approx P(\omega_l|\omega_{l-1},\mathbf{h}_{l-2},\mathbf{\Phi}) . \tag{2.20}$$

Here, $\mathbf{\Phi}$ represents the parameters of the RNN language model, comprising the weights and biases.

Similarly to an n-gram language model, the RNN language model parameters can also be trained using the Maximum Likelihood (ML) criterion of $\mathcal{F}_{\text{LM-ML}}$ in (2.18) [102]. This

can be expressed in the form of a per-word Cross-Entropy (CE) style criterion, as

$$\mathcal{F}_{\text{LM-ML}}\left(\boldsymbol{\Phi}\right) = \sum_{l=1}^{L} \log P\left(\omega_l^{\text{ref}} \middle| \omega_{l-1}^{\text{ref}}, \boldsymbol{h}_{l-2}, \boldsymbol{\Phi}\right), \tag{2.21}$$

where $\boldsymbol{h}_{l-2}$ is computed using the past words from the manual transcription, $\boldsymbol{\omega}_{1:l-2}^{\text{ref}}$. Here again, a sum over utterances is omitted for brevity. The RNN language model produces a discrete categorical distribution over words at its output. This is often implemented using a softmax output layer. NN layers, such as a softmax, are discussed in Section 2.2.5. Using a large vocabulary may necessitate having a large output layer, which may require both having many parameters and a high computational cost in computing the output softmax normalisation. One possible method to reduce the number of parameters in the output layer is to categorise words into classes [105]. The computational cost associated with the softmax normalisation can be avoided by training the outputs to naturally produce values that are close to being normalised, without having to explicitly apply a softmax normalisation. Techniques such as variance regularisation [133] and noise contrastive estimation [59] can be used for this purpose.

It can however be computationally expensive to use an RNN language model to compute the hypothesis posteriors required to perform recognition using (2.14), as the likelihood of each word depends on the full history, and therefore each unique hypothesis requires a separate sequential pass through the RNN language model [96]. One strategy to limit the computational cost when performing recognition is to limit the number of hypotheses for which language scores need to be computed, to only those captured within an n-best list [102] or a lattice [96], generated by a first pass decoding using an n-gram language model. The lattice rescoring method in [96] uses an n-gram approximation of the RNN language model, to allow the rescored hypotheses to be represented compactly as a lattice.

2.2.2 Sub-word units and dictionary

The conditional observation sequence likelihood, $p\left(\mathbf{O}_{1:T}|\boldsymbol{\omega}_{1:L}\right)$, in (2.14) relates a sequence of observations to a sequence of words. The observations are often sampled using a frame shift of around 10ms, and as such, an utterance with a duration of several seconds can have on the order of $T \approx 100$ to 1000 observation frames. On the other hand, the same utterance can be spanned by on the order of $L \approx 10$ words. It can be difficult to model the many possible alignments between these observation and word sequences that operate at drastically different length scales. One technique that is often used to facilitate this difference is to decompose

words into latent variable states, $\mathbf{s}_{1:T}$, with the same sequence length as the observations,

$$p\left(\mathbf{O}_{1:T}|\boldsymbol{\omega}_{1:L}\right) = \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}_{1:L}}} p\left(\mathbf{O}_{1:T}, \mathbf{s}_{1:T}|\boldsymbol{\omega}_{1:L}\right), \tag{2.22}$$

where $\mathcal{G}_{\boldsymbol{\omega}_{1:L}}$ is the set of all state sequences that can represent a word sequence, $\boldsymbol{\omega}_{1:L}$.

These states are often defined by first decomposing a word into sub-word units, then further decomposing the sub-word units into states. The mapping from words to sub-word unit sequences is referred to as a dictionary, which is used to define $\mathcal{G}_{\boldsymbol{\omega}_{1:L}}$. For common languages, such as English, expert knowledge is generally available to determine an appropriate set of sub-word units that have relations to the acoustic realisations. These forms of sub-word units are called phones. Possible alternate pronunciations can be captured in the dictionary by having multiple phone sequence decompositions of each word. However, the expert knowledge required to obtain a phonetic dictionary can be expensive to obtain, and may not even be available for languages with very few speakers. For many languages, it is also possible to use an orthographic sub-word unit decomposition of words [83]. These orthographic sub-word units are known as graphemes. No expert knowledge is needed to obtain this decomposition, as the words are decomposed exactly as how they are spelt. However, the closeness of the relationships between individual graphemes and phonemes can vary greatly across languages. This can lead to a greater burden on the graphemic acoustic model to capture the wider variety of acoustic realisations of individual graphemes.

One possible method to obtain a phonetic dictionary while limiting the required linguistic expertise is to use a Grapheme-to-Phoneme (G2P) mapping [29]. These either use hand-written rules [81] or train a probabilistic model to perform the mapping [129]. When using the latter method, a G2P model is trained to map from the graphemic sequence of each word to its possible phonetic sequences, using just a limited number of phonetic dictionary entries. This G2P model can then be used to predict the phonetic sequences of other words for which the phonetic decompositions are not known.

2.2.3 Hidden Markov model

The HMM is one possible model that can be used to compute the joint likelihood of the observation and state sequences, $p\left(\mathbf{O}_{1:T}, \mathbf{s}_{1:T}|\boldsymbol{\omega}_{1:L}\right)$ in (2.22) [3, 118]. By modelling this joint likelihood, the HMM can be viewed as a generative model, as it is potentially possible to sample observation and state sequence data pairs from the model. This can be used to obtain hypothesis posteriors using (2.13), (2.14), and (2.22). It is possible to train an HMM using discriminative criteria, several of which are discussed in Section 2.3.3.

The joint observation and state sequence likelihood can be factorised into

$$p\left(\mathbf{O}_{1:T}, \mathbf{s}_{1:T} | \boldsymbol{\omega}_{1:L}\right) = P\left(\mathbf{s}_{1:T} | \boldsymbol{\omega}_{1:L}\right) p\left(\mathbf{O}_{1:T} | \mathbf{s}_{1:T}, \boldsymbol{\omega}_{1:L}\right). \tag{2.23}$$

Here, $P\left(\mathbf{s}_{1:T} | \boldsymbol{\omega}_{1:L}\right)$ is referred to as the alignment model and $p\left(\mathbf{O}_{1:T} | \mathbf{s}_{1:T}, \boldsymbol{\omega}_{1:L}\right)$ is the acoustic model. In the HMM, two simplifying approximations are made to allow for tractable inference [146]. First, the probability of the current state is assumed to be conditionally independent of all other states, observations, and words, when given the previous state,

$$P\left(\mathbf{s}_{1:T} | \boldsymbol{\omega}_{1:L}\right) \approx \prod_{t=1}^{T} P\left(s_t | s_{t-1}\right) \quad , \text{ if } \mathbf{s}_{1:T} \in \mathcal{G}_{\omega_{1:L}}. \tag{2.24}$$

The dictionary defines a one-to-many mapping of the decompositions of words, such that $P\left(\mathbf{s}_{1:T} | \boldsymbol{\omega}_{1:L}\right) = 0$ if $\mathbf{s}_{1:T} \notin \mathcal{G}_{\omega_{1:L}}$. With this approximation, the alignment model, $P\left(\mathbf{s}_{1:T} | \boldsymbol{\omega}_{1:L}\right)$, is composed of the transition probabilities, $P\left(s_t | s_{t-1}\right)$. The second approximation is that the current observation is assumed to be conditionally independent of all other observations, states, and words, when given the current state,

$$p\left(\mathbf{O}_{1:T} | \mathbf{s}_{1:T}, \boldsymbol{\omega}_{1:L}\right) \approx \prod_{t=1}^{T} p\left(\mathbf{o}_t | s_t\right). \tag{2.25}$$

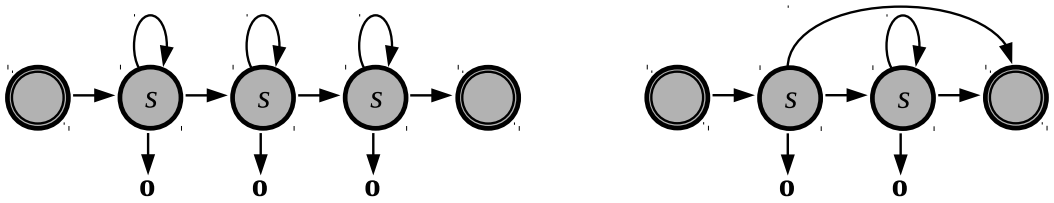
With this approximation, the acoustic model, $p\left(\mathbf{O}_{1:T} | \mathbf{s}_{1:T}, \boldsymbol{\omega}_{1:L}\right)$, is composed of observation likelihoods, $p\left(\mathbf{o}_t | s_t\right)$. Substituting these approximations into (2.23) leads to the approximate joint likelihood of

$$p\left(\mathbf{O}_{1:T}, \mathbf{s}_{1:T} | \boldsymbol{\omega}_{1:L}\right) \approx \prod_{t=1}^{T} P\left(s_t | s_{t-1}\right) p\left(\mathbf{o}_t | s_t\right) \quad , \text{ if } \mathbf{s}_{1:T} \in \mathcal{G}_{\omega_{1:L}}. \tag{2.26}$$

Using (2.13), (2.14), (2.22), and (2.26), the hypothesis posteriors needed to perform recognition can be obtained as

$$P\left(\boldsymbol{\omega}_{1:L} | \mathbf{O}_{1:T}\right) = \frac{P^{\gamma}\left(\boldsymbol{\omega}_{1:L}\right) \sum\limits_{\mathbf{s}_{1:T} \in \mathcal{G}_{\omega_{1:L}}} \prod\limits_{t=1}^{T} P^{\gamma}\left(s_t | s_{t-1}\right) p^{\kappa}\left(\mathbf{o}_t | s_t\right)}{\sum\limits_{\boldsymbol{\omega}'_{1:L'}} P^{\gamma}\left(\boldsymbol{\omega}'_{1:L'}\right) \sum\limits_{\mathbf{s}'_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}'_{1:L'}}} \prod\limits_{t=1}^{T} P^{\gamma}\left(s'_t | s'_{t-1}\right) p^{\kappa}\left(\mathbf{o}_t | s'_t\right)}. \tag{2.27}$$

Here, γ and κ are the language and acoustic scaling factors. The language and acoustic models are often trained separately from each other. These scaling factors can be used to scale these models to have comparable dynamic ranges.



(a) Traditional 3-state topology. (b) Lattice-free 2-state topology.

Fig. 2.2 Hidden Markov model topologies.

Two commonly used HMM topologies are shown in Figure 2.2. These topologies show the allowed state transitions within each sub-word unit. In ASR, the HMM topology is often restricted to only allow left-to-right state transitions, which assumes a causal nature of speech data. Figure 2.2a shows a 3-state HMM, which is often used for ASR systems that operate on a 10ms frame shift. This HMM topology forces a minimum time of 30ms per sub-word unit. In the lattice-free framework, described in Section 2.5.2, the frame shift is often increased to 30ms to reduce the computational cost during training. The 2-state HMM topology shown in Figure 2.2b can be used with this frame shift, to also allow each sub-word unit to be traversed in a minimum of 30ms [116].

2.2.4 Context dependence and state clustering

Table 2.1 Decomposition of words into context-independent and context-dependent phones. Triphones are represented here in the HTK [168] format of [previous phone]-[centre phone]+[next phone]. The phone for silence is represented as sil.

words:	the cat
context-independent phones:	th ax k ae t
context-dependent phones:	sil-th+ax th-ax+k ax-k+ae k-ae+t ae-t+sil

It has been found that the acoustic realisations of a sub-word unit are strongly influenced by the surrounding sub-word unit context [169]. However, the approximation of (2.25) in the HMM assumes that the observations corresponding to each state of a sub-word unit are conditionally independent of all other sub-word units. This assumption may hinder the ability of the model to capture the acoustic dependence on the surrounding context. The acoustic model can be modified to reflect this dependence, by using context-dependent sub-word units. An example of a decomposition of a word sequence into context-dependent phones is shown in Table 2.1. The states that compose the context-dependent sub-word units are referred to as logical context-dependent states, c. In the HMM framework, the acoustic model can be used to independently model the observation likelihoods, $p\left(\mathbf{o}_t|c_t\right)$, for each of

the logical context-dependent states, with s replaced by c in (2.25). However, the number of logical context-dependent states increases exponentially with the context window size. Independently modelling the observation likelihood of each logical context-dependent state may require an acoustic model with a large number of parameters. This can make it difficult to train the model to generalise well to unseen data, when the quantity of training data is limited. Furthermore, there may be logical context-dependent states that are never observed, regardless of the quantity of training data, because of the limited variety of sub-word unit sequences contained within the dictionary. There will therefore be no data to independently train the observation likelihoods for these logical context-dependent states.

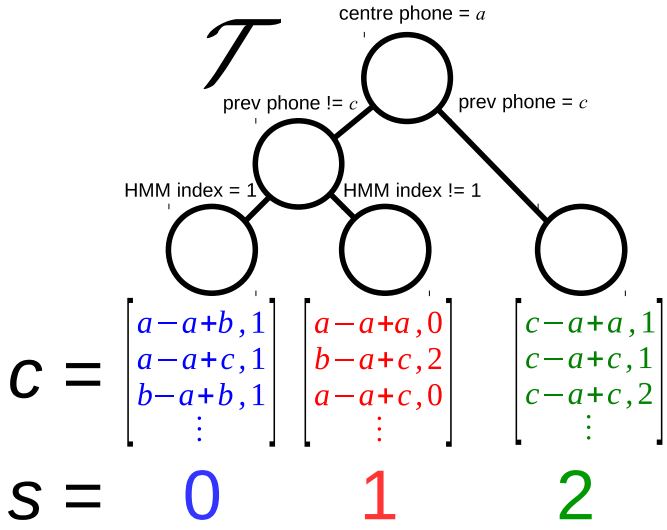


Fig. 2.3 Decision tree for state clustering, for triphone states with a centre phone of a. Each logical context-dependent state is represented as [previous phone]-[centre phone]+[next phone],[HMM state index].

One possible method to avoid having to estimate observation likelihoods for states with no data, and also to reduce the number of model parameters, is to cluster together similar logical context-dependent states, c, into state clusters, s. This is commonly done using a phonetic decision tree, shown in Figure 2.3 [169]. A decision tree represents a many-to-one mapping from logical context-dependent states to physical state clusters,

$$s_c = \mathcal{T}(c)\,. \tag{2.28}$$

The subscript in s_c is used to show that s_c is the state cluster to which logical context-dependent state c belongs to. Within each state cluster, the observation likelihoods of all logical context-dependent states are tied together to reduce the number of trainable parameters

and improve model's ability to generalise, and to avoid having to train likelihoods for which no data is available,

$$p\left(\mathbf{o}_t|c\right) = p\left(\mathbf{o}_t|s_c\right). \tag{2.29}$$

The Phonetic Decision Tree (PDT) is often trained to maximise the log-likelihood of the observations, assuming that the observation likelihoods are modelled as Gaussian probability density functions,

$$\mathcal{F}_{\text{PDT}}\left(\mathcal{T}\right) = \sum_{\mathbf{c}_{1:T}\in\mathcal{G}_{\boldsymbol{\omega}_{1:L}}} \sum_{t=1}^{T} P\left(c_t\middle|\boldsymbol{\omega}_{1:L}^{\text{ref}}, \mathbf{O}_{1:T}\right) \log \mathcal{N}\left(\mathbf{o}_t; \boldsymbol{\mu}_{\mathcal{T}(c_t)}, \boldsymbol{\Sigma}_{\mathcal{T}(c_t)}\right), \tag{2.30}$$

where $\mathcal{G}_{\boldsymbol{\omega}_{1:L}^{\text{ref}}}$ is the set of all logical context-dependent state sequences, $\mathbf{c}_{1:T}$, that can represent the manual transcription $\boldsymbol{\omega}_{1:L}^{\text{ref}}$, $P\left(c_t\middle|\boldsymbol{\omega}_{1:L}^{\text{ref}}, \mathbf{O}_{1:T}\right)$ is a distribution over the possible state alignments, and $\mathcal{N}\left(\mathbf{o}_t; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ is a Gaussian probability density function with a mean of $\boldsymbol{\mu}$ and covariance of $\boldsymbol{\Sigma}$. The Gaussian means and covariances are the empirical maximum likelihood estimates from the observations that belong to each state cluster,

$$\boldsymbol{\mu}_s = \frac{\sum_{t=1}^{T} \sum_{c:\mathcal{T}(c)=s} P\left(c_t = c\middle|\boldsymbol{\omega}_{1:L}^{\text{ref}}, \mathbf{O}_{1:T}\right) \mathbf{o}_t}{\sum_{t'=1}^{T} \sum_{c':\mathcal{T}(c')=s} P\left(c_{t'} = c'|\boldsymbol{\omega}_{1:L}^{\text{ref}}, \mathbf{O}_{1:T}\right)}, \tag{2.31}$$

and

$$\boldsymbol{\Sigma}_s = \frac{\sum_{t=1}^{T} \sum_{c:\mathcal{T}(c)=s} P\left(c_t = c\middle|\boldsymbol{\omega}_{1:L}^{\text{ref}}, \mathbf{O}_{1:T}\right) \left(\mathbf{o}_t - \boldsymbol{\mu}_s\right)\left(\mathbf{o}_t - \boldsymbol{\mu}_s\right)^{\text{T}}}{\sum_{t'=1}^{T} \sum_{c':\mathcal{T}(c')=s} P\left(c_{t'} = c'|\boldsymbol{\omega}_{1:L}^{\text{ref}}, \mathbf{O}_{1:T}\right)}. \tag{2.32}$$

However, finding the globally optimal decision tree is computationally intractable [72]. As such, the decision tree is often trained by following an iterative processes that chooses the greedy split at each iteration, with the largest increase in log-likelihood, from a set of phonetically motivated questions [169]. Examples of such questions are, *does the next phone belong to the set* $\{\cdots\}$, or *is the HMM state index equal to* i. At each iteration, v, the decision tree can be updated as

$$\mathcal{T}^{(v+1)} = \arg\max_{\mathcal{T}^{(v)+1}} \left\{\mathcal{F}_{\text{PDT}}\left(\mathcal{T}^{(v)+1}\right) - \mathcal{F}_{\text{PDT}}\left(\mathcal{T}^{(v)}\right)\right\}, \tag{2.33}$$

where $\mathcal{T}^{(v)+1}$ are the possible decision trees with one more split than $\mathcal{T}^{(v)}$, as is illustrated in Figure 2.4. However, this method of training the decision tree is not guaranteed to result in a solution that is globally optimal over the whole tree.

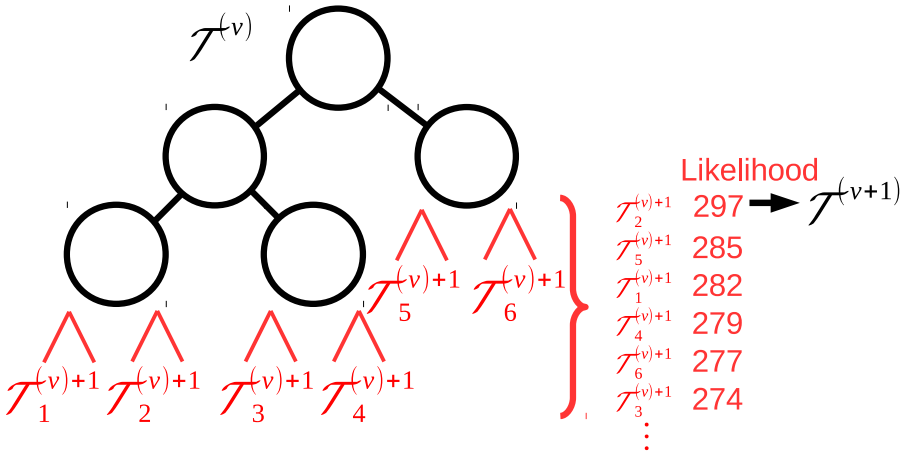


Fig. 2.4 Training a decision tree by selecting the greedy split at each iteration. At each iteration, all possible next splits, $\mathcal{T}_i^{(v)+1}$, are listed in order of likelihood, and the most likely split is chosen as the next split, $\mathcal{T}^{(v+1)}$. Only a single tree root is shown here, but in practice, the most likely split is chosen over all roots.

Often, separate decision tree roots are used for different centre phones or HMM state indexes. This enforces a prior knowledge that logical context-dependent states belonging to different tree roots have significantly different acoustic realisations.

2.2.5 Acoustic model

The acoustic model is used to compute $p\left(\mathbf{O}_{1:T}|\mathbf{s}_{1:T}, \boldsymbol{\omega}_{1:L}\right)$ in (2.23). Under the HMM approximation of (2.25), this requires computing the observation likelihoods for each frame, $p\left(\mathbf{o}_t|s\right)$.

Gaussian mixture model

One possible method is to use a Gaussian Mixture Model (GMM) [76],

$$p\left(\mathbf{o}_t|s, \boldsymbol{\Phi}\right) = \sum_{m=1}^{M_s} \varsigma_{sm} \mathcal{N}\left(\mathbf{o}_t; \boldsymbol{\mu}_{sm}, \boldsymbol{\Sigma}_{sm}\right), \tag{2.34}$$

where $\boldsymbol{\Phi} = \{\boldsymbol{\mu}_{sm}, \boldsymbol{\Sigma}_{sm}, \varsigma_{sm} \quad \forall s, m\}$ are the GMM parameters, consisting of the means, $\boldsymbol{\mu}_{sm}$, covariance matrices, $\boldsymbol{\Sigma}_{sm}$, and component priors, ς_{sm}, that satisfy $\varsigma_{sm} \geq 0$ and $\sum_m \varsigma_{sm} = 1$. Here, m is the mixture component index and M_s is the number of Gaussian mixture components in state cluster s. The GMM can be viewed as a generative model, as observations can be sampled from it. The bulk of the trainable parameters in the GMM reside within the covariance matrices, $\boldsymbol{\Sigma}_{sm}$. To reduce the number of trainable parameters and improve

model's ability to generalise, it is common to force $\boldsymbol{\Sigma}_{sm}$ to be diagonal. Using diagonal covariance matrices assumes that within each Gaussian mixture component, the observation feature dimensions are uncorrelated. This assumption is relaxed by having multiple Gaussian mixture components for each state cluster in a GMM, as multiple diagonal components can be used together to model correlated feature dimensions.

Feed-forward deep neural network

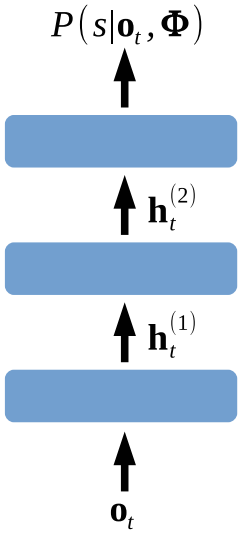


Fig. 2.5 Feed-forward deep neural network. Each rectangular block here represents a linear transformation of (2.36), followed by a nonlinear operation of (2.37).

Recent advances in ASR have shown that using a Neural Network (NN) classifier as the acoustic model can yield a performance that is often superior to that of a GMM [67]. Using an NN as the acoustic model to compute the observation likelihoods, $p(\mathbf{o}_t|s)$, required for the HMM in (2.26) is referred to as a hybrid system [12]. The feed-forward Deep NN (DNN), illustrated in Figure 2.5, consists of multiple layers of nonlinear transformations of the input observation features,

$$\mathbf{h}_t^{(0)} = \mathbf{o}_t \tag{2.35}$$

$$\mathbf{z}_t^{(k)} = \mathbf{W}^{(k)}\mathbf{h}_t^{(k-1)} + \mathbf{b}^{(k)} \tag{2.36}$$

$$\mathbf{h}_t^{(k)} = \mathbf{g}^{(k)}\left(\mathbf{z}_t^{(k)}\right) \tag{2.37}$$

$$\mathbf{y}_t = \mathbf{g}^{\text{softmax}}\left(\mathbf{z}_t^{(K+1)}\right), \tag{2.38}$$

where k is the layer index, K is the number of hidden layers, $\mathbf{W}^{(k)}$ are the weight matrices, $\mathbf{b}^{(k)}$ are the bias vectors, $\mathbf{h}_t^{(k)}$ are the hidden activations, $\mathbf{z}_t^{(k)}$ are referred to as the pre-

nonlinearity activations, and $\mathbf{y}_t$ are the outputs. The activation functions for each layer, $\mathbf{g}^{(k)}$, are element-wise nonlinear functions, often in the form of a tanh, sigmoid,

$$g_i^{\text{sigmoid}}(\mathbf{z}) = \frac{1}{1+\exp(-z_i)}, \tag{2.39}$$

or Rectified Linear Unit (ReLU) [51],

$$g_i^{\text{ReLU}}(\mathbf{z}) = \max\{0, z_i\}. \tag{2.40}$$

The output layer of the NN uses a softmax activation function,

$$g_i^{\text{softmax}}(\mathbf{z}) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}. \tag{2.41}$$

The NN outputs form a categorical distribution, which can be interpreted as a state cluster posterior distribution,

$$P(s|\mathbf{o}_t, \mathbf{\Phi}) = y_{ts}, \tag{2.42}$$

where here, $\mathbf{\Phi} = \left\{\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \cdots, \mathbf{W}^{(K+1)}, \mathbf{b}^{(K+1)}\right\}$ represents the set of NN model parameters. The NN therefore can be viewed as a discriminative model that can be used to classify each observation frame into the set of state clusters.

The HMM requires the computation of observation likelihoods. This can be expressed in terms of the state cluster posteriors as

$$p(\mathbf{o}_t|s) = \frac{P(s|\mathbf{o}_t)\, p(\mathbf{o}_t)}{P(s)}. \tag{2.43}$$

In the hybrid NN-HMM framework [12], the state cluster posteriors from the output of the NN, $P(s|\mathbf{o}_t, \mathbf{\Phi})$, are used to compute these observation likelihoods. The state cluster priors can be estimated from the NN state cluster posteriors as

$$P(s) = \int P(s|\mathbf{o}, \mathbf{\Phi})\, p(\mathbf{o})\, d\mathbf{o} \tag{2.44}$$

$$\approx \sum_{t=1}^{T} \frac{1}{T} P(s|\mathbf{o}_t, \mathbf{\Phi}). \tag{2.45}$$

The observation priors, $p(\mathbf{o}_t)$, do not have an impact on the hypothesis posteriors, as they cancel out in the numerator and denominator of the hypothesis posteriors, when substituting (2.43) into (2.27).

As such, when using an NN acoustic model, normalised hypothesis posteriors can still be obtained by omitting the observation priors in (2.43) and computing a scaled observation likelihood,

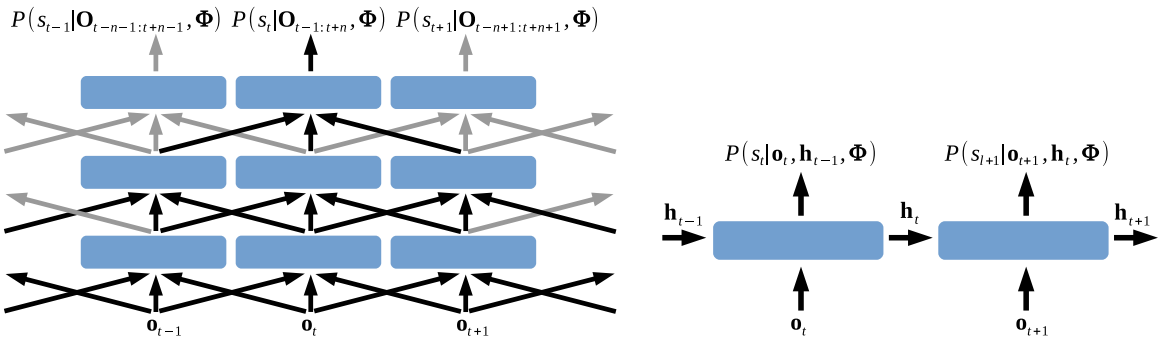
$$\mathcal{A}\left(\mathbf{o}_t, s, \mathbf{\Phi}\right) = \frac{P\left(s|\mathbf{o}_t, \mathbf{\Phi}\right)}{P\left(s\right)}. \tag{2.46}$$

This scaled likelihood does not represent a valid probability density function, as it will only be normalised when multiplied by a valid observation prior. Using this scaled observation likelihood, the hypothesis posteriors of (2.27) for the hybrid NN-HMM system can be expressed as

$$P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T}, \mathbf{\Phi}\right) = \frac{P^{\gamma}\left(\boldsymbol{\omega}_{1:L}\right) \sum\limits_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}_{1:L}}} \prod\limits_{t=1}^{T} P^{\gamma}\left(s_t|s_{t-1}\right) \mathcal{A}^{\kappa}\left(\mathbf{o}_t, s_t, \mathbf{\Phi}\right)}{\sum\limits_{\boldsymbol{\omega}'_{1:L'}} P^{\gamma}\left(\boldsymbol{\omega}'_{1:L'}\right) \sum\limits_{\mathbf{s}'_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}'_{1:L'}}} \prod\limits_{t=1}^{T} P^{\gamma}\left(s'_t|s'_{t-1}\right) \mathcal{A}^{\kappa}\left(\mathbf{o}_t, s'_t, \mathbf{\Phi}\right)}. \tag{2.47}$$

One disadvantage of the GMM is that every state cluster, s, has its own set of parameters, $\{\boldsymbol{\mu}_{sm}, \mathbf{\Sigma}_{sm}, \varsigma_{sm} \quad \forall m\}$. This can result in a large number of trainable parameters and poor model generalisation when the number of state clusters is large. Furthermore, there may be context-dependent phones that are never observed in the data, because some phone sequences may never occur in the dictionary. The NN overcomes this issue, as all of the parameters in an NN, up to the final hidden layer, are shared across all state clusters.

Neural network temporal context



(a) Time-delay neural network. Dark arrows represent the flow of information used to compute the current output.

(b) Recurrent neural network. Information about the past history of observations is captured in $\mathbf{h}_l$.

Fig. 2.6 Neural network topologies that capture extended temporal contexts.

It has been shown that performance improvements can be obtained by using NN topologies that take into account wider temporal contexts of the observations [145]. Two such NN topologies are the Time-Delay NN (TDNN) [147] and the RNN [121], illustrated in Figures 2.6a and 2.6b respectively. In a standard feed-forward DNN, a limited temporal context is often captured by splicing consecutive frames of observations together to form the input [12]. The TDNN generalises this concept by performing temporal splicing at each hidden layer. This replaces (2.36) in the DNN feed-forward operation with

$$\mathbf{z}_t^{(k)} = \sum_{\tau=-n}^{n} \mathbf{W}_\tau^{(k)} \mathbf{h}_{t+\tau}^{(k-1)} + \mathbf{b}^{(k)}. \tag{2.48}$$

Dilation can be applied to the splice window to limit the increase in the number of parameters as the context window size increases [112].

The feed-forward DNN with a spliced input and the TDNN are both only able to capture a finite temporal context. The RNN can take into account the full history of past observations when classifying each frame. As such, the RNN outputs can be interpreted as state cluster posteriors, given the current and past observations,

$$P\left(s_t|\mathbf{O}_{1:t}\right) \approx P\left(s_t|\mathbf{o}_t, \mathbf{h}_{t-1}, \mathbf{\Phi}\right). \tag{2.49}$$

Here, a fixed-dimensional representation, $\mathbf{h}_{t-1}$, is used to capture all past observations, $\mathbf{O}_{1:t-1}$. Although the RNN can potentially capture the full past history of observations, the computation of each posterior requires the past observation frames to be processed sequentially. This can make it difficult to parallelise the computations and perform mini-batching during training. As such, only a finite past context of observations is often used during training. It is also possible to incorporate a limited future context into the RNN, by introducing a fixed delay between the input and output.

One popular RNN topology is the Long Short-Term Memory (LSTM) [70]. In this thesis, the LSTM topology proposed in [125] is used, and is illustrated in Figure 2.7. A single LSTM layer using this topology, that takes $\mathbf{o}_t$ as input and produces $\mathbf{y}_t$ as output, is described

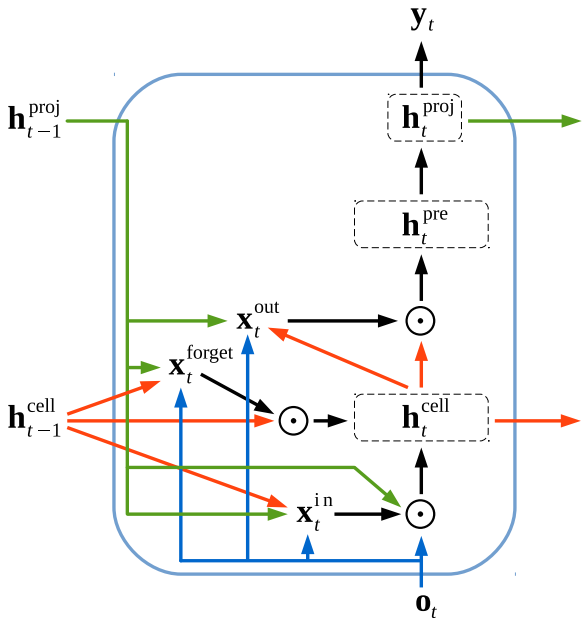


Fig. 2.7 Long short-term memory layer, with low-rank matrix factorisation projection. The arrows show how the computation of each variable is dependent on other variables. Arrows of each colour show the information flow out of different variables.

as

$$\mathbf{x}_t^{\text{in}} = \mathbf{g}^{\text{in}}\left(\mathbf{W}^{\text{ii}}\mathbf{o}_t + \mathbf{W}^{\text{ip}}\mathbf{h}_{t-1}^{\text{proj}} + \mathbf{W}^{\text{ic}}\mathbf{h}_{t-1}^{\text{cell}} + \mathbf{b}^{\text{i}}\right) \tag{2.50}$$

$$\mathbf{x}_t^{\text{forget}} = \mathbf{g}^{\text{forget}}\left(\mathbf{W}^{\text{fi}}\mathbf{o}_t + \mathbf{W}^{\text{fp}}\mathbf{h}_{t-1}^{\text{proj}} + \mathbf{W}^{\text{fc}}\mathbf{h}_{t-1}^{\text{cell}} + \mathbf{b}^{\text{f}}\right) \tag{2.51}$$

$$\mathbf{h}_t^{\text{cell}} = \mathbf{x}_t^{\text{forget}} \odot \mathbf{h}_{t-1}^{\text{cell}} + \mathbf{x}_t^{\text{in}} \odot \mathbf{g}^{\text{cell}}\left(\mathbf{W}^{\text{ci}}\mathbf{o}_t + \mathbf{W}^{\text{cp}}\mathbf{h}_{t-1}^{\text{proj}} + \mathbf{b}^{\text{p}}\right) \tag{2.52}$$

$$\mathbf{x}_t^{\text{out}} = \mathbf{g}^{\text{out}}\left(\mathbf{W}^{\text{oi}}\mathbf{o}_t + \mathbf{W}^{\text{op}}\mathbf{h}_{t-1}^{\text{proj}} + \mathbf{W}^{\text{oc}}\mathbf{h}_t^{\text{cell}} + \mathbf{b}^{\text{o}}\right) \tag{2.53}$$

$$\mathbf{h}_t^{\text{pre}} = \mathbf{x}_t^{\text{out}} \odot \mathbf{g}^{\text{pre}}\left(\mathbf{h}_t^{\text{cell}}\right) \tag{2.54}$$

$$\mathbf{h}_t^{\text{proj}} = \mathbf{W}^{\text{proj}}\mathbf{h}_t^{\text{pre}} \tag{2.55}$$

$$\mathbf{y}_t = \mathbf{g}^{\text{output}}\left(\mathbf{W}^{\text{output}}\mathbf{h}_t^{\text{proj}} + \mathbf{b}^{\text{output}}\right). \tag{2.56}$$

Here, $\mathbf{x}_t^{\text{in}}$, $\mathbf{x}_t^{\text{forget}}$, and $\mathbf{x}_t^{\text{out}}$, are referred to as the input, forget, and output gates respectively. These are used to control the flow of information through the LSTM layer, by taking the element-wise product, $\odot$, of the gates with the activations. The parameter matrices of $\mathbf{W}^{\text{ic}}$, $\mathbf{W}^{\text{fc}}$, and $\mathbf{W}^{\text{oc}}$ are referred to as peephole connections, and are constrained to be diagonal, to reduce the number of parameters. The memory cell, $\mathbf{h}_t^{\text{cell}}$, is where the recurrent memory is stored. The memory cell and gating topology of the LSTM is designed to mitigate the

occurrence of vanishing and exploding gradients during training [111], thereby improving the ability of the model to learn long span temporal dependencies in the data.

The LSTM can potentially have many parameters, from the multiple matrices and biases. This can make it difficult to train the LSTM to generalise well, when the quantity of training data is limited. In the LSTM topology used here, the activations of $\mathbf{h}_t^{\text{pre}}$ are linearly projected to $\mathbf{h}_t^{\text{proj}}$, such that $\dim\left(\mathbf{h}_t^{\text{proj}}\right) < \dim\left(\mathbf{h}_t^{\text{pre}}\right)$ [125]. This reduces the number of parameters in the LSTM layer, and can be interpreted as a low-rank matrix factorisation, discussed in Section 4.1. Although only a single LSTM layer is illustrated here, it is possible to stack multiple LSTM layers together, to increase the capacity of the model.

A single RNN acoustic model can capture the temporal context of observations trailing either backward or forward in time. It is possible to capture the complete backward and forward temporal contexts, by using two RNNs in parallel, one running forward and the other running backward in time. When used with the LSTM acoustic model topology, this is referred to as the Bi-directional LSTM (BLSTM) [57].

The use of extended temporal contexts in the acoustic model allows the conditional independence assumption of (2.25) to be relaxed. When using an RNN acoustic model running forward in time, the approximation of (2.25) can be relaxed to

$$p\left(\mathbf{O}_{1:T}|\mathbf{s}_{1:T}, \boldsymbol{\omega}_{1:L}\right) \approx \prod_{t=1}^{T} p\left(\mathbf{o}_t|\mathbf{O}_{1:t-1}, s_t\right), \tag{2.57}$$

where now the conditional dependence of the current observation on past observations is explicitly captured. Here, it is still assumed that the current observation is conditionally independent of all other words, state clusters, and future observations, when given the current state cluster and past observations. The observation likelihood can then be expressed as

$$p\left(\mathbf{o}_t|\mathbf{O}_{1:t-1}, s_t\right) = \frac{P\left(s_t|\mathbf{O}_{1:t}\right) p\left(\mathbf{o}_t|\mathbf{O}_{1:t-1}\right)}{P\left(s_t|\mathbf{O}_{1:t-1}\right)}. \tag{2.58}$$

Similarly to (2.43), the observation priors, $p\left(\mathbf{o}_t|\mathbf{O}_{1:t-1}\right)$, are independent of the hypothesis, and will not affect the resulting hypothesis posteriors. It is therefore often omitted. When using an RNN acoustic model together with an HMM, an addition approximation is often made that

$$P\left(s_t|\mathbf{O}_{1:t-1}\right) \approx P\left(s_t\right). \tag{2.59}$$

Although this approximation is difficult to justify theoretically, it eliminates the need to separately model $P\left(s_t|\mathbf{O}_{1:t-1}\right)$, and often results in a reasonable performance in practice [121]. Using these approximations, scaled observation likelihoods can be obtained from the

RNN state cluster posteriors using

$$\mathcal{A}\left(\mathbf{o}_t, \mathbf{O}_{1:t-1}, s_t, \boldsymbol{\Phi}\right) = \frac{P\left(s_t|\mathbf{o}_t, \mathbf{h}_{t-1}, \boldsymbol{\Phi}\right)}{P\left(s_t\right)}. \tag{2.60}$$

As with (2.46), these scaled observation likelihoods also do not represent valid probability density functions. By using an RNN acoustic model together with an HMM in the hybrid framework, hypothesis posteriors of (2.27) can be computed as

$$P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right) = \frac{P^{\gamma}\left(\boldsymbol{\omega}_{1:L}\right) \sum\limits_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}_{1:L}}} \prod\limits_{t=1}^{T} P^{\gamma}\left(s_t|s_{t-1}\right) \mathcal{A}^{\kappa}\left(\mathbf{o}_t, \mathbf{O}_{1:t-1}, s_t, \boldsymbol{\Phi}\right)}{\sum\limits_{\boldsymbol{\omega}'_{1:L'}} P^{\gamma}\left(\boldsymbol{\omega}'_{1:L'}\right) \sum\limits_{\mathbf{s}'_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}'_{1:L'}}} \prod\limits_{t=1}^{T} P^{\gamma}\left(s'_t|s'_{t-1}\right) \mathcal{A}^{\kappa}\left(\mathbf{o}_t, \mathbf{O}_{1:t-1}, s'_t, \boldsymbol{\Phi}\right)}. \tag{2.61}$$

2.2.6 Feature extraction

It has thus far been assumed that inputs are available in the form of per-frame feature vectors, $\mathbf{o}_t$, which together form the observation sequence,

$$\mathbf{O}_{1:T} = \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T. \tag{2.62}$$

These features are often extracted from the an underlying raw audio waveform of the audio speech signal,

$$\mathbf{x}_{1:\tau} = x_1, x_2, \cdots, x_\tau, \tag{2.63}$$

through the feature extractor,

$$\mathbf{o}_t = \boldsymbol{\varphi}_t\left(\mathbf{x}_{1:\tau}\right), \tag{2.64}$$

where τ is the length of the raw audio waveform. Although it is possible to construct an ASR system that operates directly on the raw audio waveform as inputs [110], the extracted features can have a more compact representation of the discriminative information, which may be more useful for recognition.

Hand-crafted features

The hand-crafted filterbank, Mel-Frequency Cepstral Coefficients (MFCC) [101], and Perceptual Linear Predictive (PLP) [64] features are common choices of input representations. These are extracted by first segmenting the audio stream into overlapping 25ms windows, with a 10ms shift between consecutive frames. A Discrete Fourier Transform (DFT) is then

applied to each window. Only the magnitude spectra are retained. The magnitude spectra are then down-sampled by integrating across triangular Mel-scale frequency bins. This results in filterbank features.

As is described in Section 2.2.5, diagonal covariance matrices are often used in a GMM-HMM system, to reduce the number of model parameters and improve the ability to generalise. This assumes some degree of independence between the feature vector dimensions. One possible method to improve the independence between the feature dimensions is through a Discrete Cosine Transform (DCT) of the filterbank features. The MFCC features are formed by only retaining the first few DCT dimensions. However, this truncation may result in information loss. Therefore when using acoustic models that do not assume independence between feature dimensions, the filterbank features may be preferred.

For PLP, the magnitude spectra are downsampled by integrating across critical bank filters. These filters are traditionally spaced along the Bark scale. However, in this thesis, the Mel scale is again used. The features are then scaled using the equal-loudness pre-emphasis curve and intensity-loudness power law. Auto-correlation coefficients that represent the features are then obtained and truncated. Finally, an inverse DFT is applied to the log-spectrum of the auto-correlation coefficients, and truncated, to produce the PLP features.

There are also many other hand-crafted feature types, such as those relying on gammatone filters [126], and those operating on the DFT phase spectrum [128]. The extracted features are often made more robust to environment and speaker variations through mean and variance normalisation. More powerful normalisation techniques, such as Constrained Maximum Likelihood Linear Regression (CMLLR) [43] and vocal tract length normalisation [170] can also be used. Finally, it has been found that when using non-recurrent models, performance improvements can often be obtained by introducing a finite temporal context into the features. This can be done by concatenating together the features with their temporal derivatives, or by splicing together a window of consecutive frames.

Neural network bottleneck features

Many of these hand-crafted features are inspired by insights from signal processing and human audio perception. However, these may not be the most optimal for ASR. It may be more appropriate to learn a feature extraction from the data. An NN provides a flexible model that can be trained as a feature extractor [65],

$$\acute{\mathbf{o}}_t = \boldsymbol{\varphi}_t\left(\mathbf{O}_{1:T}|\boldsymbol{\Phi}\right), \tag{2.65}$$

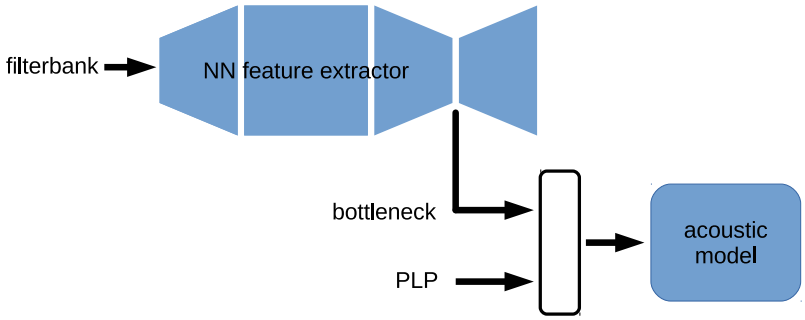


Fig. 2.8 Using an NN bottleneck feature extractor in tandem with an acoustic model. PLP features are concatenated together with the bottleneck features to provide additional information. The acoustic model can be an NN or a GMM.

where Φ here represents the parameters of an NN feature extractor. The feature extractor, φ, takes in input features, $\mathbf{O}_{1:T}$, which may be hand-crafted features or the raw audio waveform, and produces the extracted features, $\acute{\mathbf{o}}_t$, which together form the extracted observation sequence, $\acute{\mathbf{O}}_{1:T} = \acute{\mathbf{o}}_1, \cdots, \acute{\mathbf{o}}_T$. It has been found that compact representations can be obtained by placing a bottleneck, having a small number of nodes, at one of the layers along the NN topology, and using the bottleneck layer activations as the extracted features [58]. The NN can be trained in an autoencoder [68] or in a discriminative [65] manner. It is also possible to improve noise robustness, by training the feature extractor to predict clean features from noisy input features [144]. The bottleneck features can be used together with hand-crafted features, referred to as tandem features, to leverage upon the advantages of multiple representations [148], as is shown in Figure 2.8. The use of NN bottleneck features as inputs to a GMM-HMM system is referred to as a tandem system.

However, the ability of the NN feature extractor to generalise depends on the quantity of training data. It is possible to import additional data from other tasks, such as from other languages, forming what is known as multi-lingual bottleneck features [36]. This makes the assumption that the features from the related tasks share similar discriminative representations of information.

2.3 Training criteria

The parameters of the models first need to be trained, before the systems can be used for recognition. Training refers to the process of finding a set of model parameters, Φ^*, that are

optimal[2] with respect to a training criterion, $\mathcal{F}$, computed over the training data, $\mathcal{D}$,

$$\boldsymbol{\Phi}^* = \arg\max_{\boldsymbol{\Phi}} \mathcal{F}\left(\boldsymbol{\Phi}\right)\big|_{\mathcal{D}}\,. \tag{2.66}$$

This section reviews several possible methods that can be used to train GMM and NN acoustic models, and the HMM alignment model.

2.3.1 Maximum likelihood

The HMM is a generative model of the joint distribution between the observation and state cluster sequences, $p\left(\mathbf{O}_{1:T}, \mathbf{s}_{1:T}|\boldsymbol{\omega}\right)$. A GMM can be used to compute the HMM observation likelihoods using (2.34). One method to train a GMM-HMM is to maximise the log-likelihood of the training data, which consists of observation sequence and manual transcription pairs, $\left\{\mathbf{O}_{1:T}, \boldsymbol{\omega}^{\text{ref}}\right\}$, using the Maximum Likelihood (ML) criterion of

$$\begin{aligned}\mathcal{F}_{\text{ML}}\left(\boldsymbol{\Phi}\right) &= \log p\left(\mathbf{O}_{1:T}, \boldsymbol{\omega}^{\text{ref}}\middle|\boldsymbol{\Phi}\right) &(2.67)\\ &= \log p\left(\mathbf{O}_{1:T}\middle|\boldsymbol{\omega}^{\text{ref}}, \boldsymbol{\Phi}\right) + \log P\left(\boldsymbol{\omega}^{\text{ref}}\right), &(2.68)\end{aligned}$$

where here, $\boldsymbol{\Phi}$ represents the GMM parameters and the HMM transition probabilities. A sum over utterances in the training data is omitted for brevity. The GMM is often trained separately from the language model. As such, the criterion can be simplified to

$$\mathcal{F}_{\text{ML}}\left(\boldsymbol{\Phi}\right) = \log p\left(\mathbf{O}_{1:T}\middle|\boldsymbol{\omega}^{\text{ref}}, \boldsymbol{\Phi}\right). \tag{2.69}$$

The likelihood requires the marginalisation over the possible state cluster sequences,

$$\mathcal{F}_{\text{ML}}\left(\boldsymbol{\Phi}\right) = \log \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}^{\text{ref}}}} p\left(\mathbf{s}_{1:T}, \mathbf{O}_{1:T}\middle|\boldsymbol{\omega}^{\text{ref}}, \boldsymbol{\Phi}\right), \tag{2.70}$$

which can be treated as latent variables. This can be difficult to optimise directly. As such, the criterion can be optimised indirectly, by maximising an evidence lower bound to the log-likelihood, using the Expectation Maximisation (EM) algorithm [32]. This is also known as the Baum-Welch algorithm [4] when applied to HMMs, and is a variant of variational

[2]Optimisation can refer to either a maximisation or minimisation, depending on the criterion. Maximisation can be changed to minimisation, simply by negating the criterion.

inference, which is described in more detail in Section 3.1.2. The evidence lower bound is,

$$\mathcal{F}_{\text{EM}}\left(\boldsymbol{\Phi}, Q\right) = \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}^{\text{ref}}}} Q\left(\mathbf{s}_{1:T} \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}\right) \log \frac{p\left(\mathbf{s}_{1:T}, \mathbf{O}_{1:T} \middle| \boldsymbol{\omega}^{\text{ref}}, \boldsymbol{\Phi}\right)}{Q\left(\mathbf{s}_{1:T} | \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}\right)} \tag{2.71}$$

$$= \log p\left(\mathbf{O}_{1:T} \middle| \boldsymbol{\omega}^{\text{ref}}, \boldsymbol{\Phi}\right) - \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}^{\text{ref}}}} Q\left(\mathbf{s}_{1:T} \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}\right) \log \frac{Q\left(\mathbf{s}_{1:T} \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}\right)}{P\left(\mathbf{s}_{1:T} | \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)}, \tag{2.72}$$

where $Q\left(\mathbf{s}_{1:T} \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}\right)$ is an approximate distribution over the state cluster sequence. The EM algorithm performs optimisation jointly over both $Q\left(\mathbf{s}_{1:T} \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}\right)$ and $\boldsymbol{\Phi}$. The term on the left of (2.72) is the log-likelihood of (2.69), while the term on the right of (2.72) is a Kullback-Leibler (KL)-divergence between the approximate distribution and the one given by the model, which is non-negative. As such, $\mathcal{F}_{\text{EM}} \leq \mathcal{F}_{\text{ML}}$, and maximising $\mathcal{F}_{\text{EM}}$ maximises a lower bound to $\mathcal{F}_{\text{ML}}$.

The EM algorithm trains the model by iterating over the following two steps.

1. **The Expectation step** maximises $\mathcal{F}_{\text{EM}}\left(\boldsymbol{\Phi}, Q\right)$ in (2.72) with respect to Q, keeping the parameters fixed at the values from the previous training iteration, $\boldsymbol{\Phi}^{(v-1)}$. This is achieved by setting the approximate distribution to the distribution over state cluster sequences of the previous model iteration,

$$Q\left(\mathbf{s}_{1:T} \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}\right) = P\left(\mathbf{s}_{1:T} \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}^{(v-1)}\right). \tag{2.73}$$

   This minimises the KL-divergence in (2.72), thereby minimising the difference between $\mathcal{F}_{\text{EM}}$ and $\mathcal{F}_{\text{ML}}$, and leading to $\mathcal{F}_{\text{EM}} = \mathcal{F}_{\text{ML}}$.

2. **The Maximisation step** maximises $\mathcal{F}_{\text{EM}}\left(\boldsymbol{\Phi}, Q\right)$ in (2.71) with respect to $\boldsymbol{\Phi}$, with the Lagrange multiplier constraints to ensure that $\sum_{s_t} P\left(s_t | s_{t-1}\right) = 1$ and $\sum_m \varsigma_{ms} = 1$. Equating the criterion derivatives to zero, $\frac{\partial \mathcal{F}_{\text{EM}}}{\partial \boldsymbol{\Phi}} = 0$, leads to analytical solutions for

the model parameters [76],

$$P^{(v)}\left(s_\tau|s_{\tau-1}\right) = \frac{\sum_{t=1}^{T} P\left(s_t = s_\tau, s_{t-1} = s_{\tau-1}\middle|\mathbf{O}_{1:T}, \boldsymbol{\Phi}^{(v-1)}\right)}{\sum_{t'=1}^{T}\sum_{s'\in\mathcal{T}} P\left(s_{t'} = s', s_{t'-1} = s_{\tau-1}|\mathbf{O}_{1:T}, \boldsymbol{\Phi}^{(v-1)}\right)} \tag{2.74}$$

$$\varsigma_{ms}^{(v)} = \frac{\sum_{t=1}^{T} P\left(s_t = s, m_t = m\middle|\mathbf{O}_{1:T}, \boldsymbol{\Phi}^{(v-1)}\right)}{\sum_{t'=1}^{T}\sum_{m'=1}^{M_s} P\left(s_{t'} = s, m_{t'} = m'|\mathbf{O}_{1:T}, \boldsymbol{\Phi}^{(v-1)}\right)} \tag{2.75}$$

$$\boldsymbol{\mu}_{ms}^{(v)} = \frac{\sum_{t=1}^{T} P\left(s_t = s, m_t = m\middle|\mathbf{O}_{1:T}, \boldsymbol{\Phi}^{(v-1)}\right)\mathbf{o}_t}{\sum_{t'=1}^{T} P\left(s_{t'} = s, m_{t'} = m|\mathbf{O}_{1:T}, \boldsymbol{\Phi}^{(v-1)}\right)} \tag{2.76}$$

$$\boldsymbol{\Sigma}_{ms}^{(v)} = \frac{\sum_{t=1}^{T} P\left(s_t = s, m_t = m\middle|\mathbf{O}_{1:T}, \boldsymbol{\Phi}^{(v-1)}\right)\left(\mathbf{o}_t - \boldsymbol{\mu}_{ms}^{(v)}\right)\left(\mathbf{o}_t - \boldsymbol{\mu}_{ms}^{(v)}\right)^{\mathrm{T}}}{\sum_{t'=1}^{T} P\left(s_{t'} = s, m_{t'} = m|\mathbf{O}_{1:T}, \boldsymbol{\Phi}^{(v-1)}\right)}. \tag{2.77}$$

Here, $\sum_{s\in\mathcal{T}}$ sums over all state clusters at the leaves of the decision tree, $\mathcal{T}$. The EM algorithm alternates between optimising with respect to $\boldsymbol{\Phi}$ and $Q\left(\mathbf{s}_{1:T}\middle|\boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}\right)$, and can therefore be seen as a form of coordinate ascent.

The EM algorithm described by (2.73) to (2.77) can be computationally expensive to perform, as all possible state cluster time alignments and alternative pronunciations need to be considered. One method to reduce this computational cost is to set the approximate distribution to the 1-best state cluster alignment in the expectation step [168], replacing (2.73) with

$$Q\left(\mathbf{s}_{1:T}\middle|\boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}\right) = \delta\left(\mathbf{s}_{1:T}, \mathbf{s}_{1:T}^{(v-1)*}\right), \tag{2.78}$$

where

$$\mathbf{s}_{1:T}^{(v-1)*} = \arg\max_{\mathbf{s}_{1:T}} P\left(\mathbf{s}_{1:T}\middle|\boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}^{(v-1)}\right). \tag{2.79}$$

2.3.2 Frame level

Section 2.2.5 describes how in the hybrid framework, an NN can be used in place of a GMM, to compute the scaled observation likelihoods of an HMM, using (2.46). One commonly used procedure [12] to train this system is to first train the HMM transition probabilities in a GMM-HMM system using the EM algorithm, described in Section 2.3.1. 1-best state cluster alignments of the manual transcription, referred to as forced alignments, are obtained from

the GMM-HMM system. The acoustic model is then replaced by an NN. One method to train the NN is using the Cross-Entropy (CE) criterion, which maximises the conditional log-likelihood of the forced alignments [12],

$$\mathcal{F}_{\text{CE}}\left(\boldsymbol{\Phi}\right) = \sum_{t=1}^{T} \log P\left(s_t^{\text{ref}} \middle| \mathbf{o}_t, \boldsymbol{\Phi}\right), \tag{2.80}$$

where s_t^{ref} is a cluster state in the forced alignment and $\boldsymbol{\Phi}$ now represents the NN parameters. The cross-entropy criterion is commonly used in many classification tasks in machine learning. This criterion is discriminative in nature, as it aims to maximise the conditional likelihood of the correct class. When applied to training an NN in the hybrid NN-HMM system, this is similar to performing a final maximisation step in the EM algorithm, using the 1-best state cluster alignment approximation of (2.78), obtained from a GMM-HMM system.

However, training the NN toward these forced alignment targets may bias its behaviour, discouraging behaviours that favour alternative time alignments and pronunciations that can represent the manual transcriptions. Frame-level training also does not take into account the sequential nature of speech data, or the alignment and language models. Furthermore, the performance of the system is often assessed based on its word sequence hypotheses, and not its per-frame classifications. The following section reviews several criteria that take sequence-level information into account.

2.3.3 Sequence discriminative training

Sequence discriminative training methods aim to overcome the limitations of frame-level training. One possible sequence discriminative criterion is the Maximum Mutual Information (MMI) criterion [1], which maximises the conditional log-likelihood of the manual transcriptions,

$$\mathcal{F}_{\text{MMI}}\left(\boldsymbol{\Phi}\right) = \log P\left(\boldsymbol{\omega}^{\text{ref}} \middle| \mathbf{O}_{1:T}, \boldsymbol{\Phi}\right). \tag{2.81}$$

Here again, a sum over utterances is omitted for brevity. Similarly to the $\mathcal{F}_{\text{CE}}$ criterion, the $\mathcal{F}_{\text{MMI}}$ criterion is also discriminative in nature, as it aims to maximise the conditional likelihood of the correct class.

The $\mathcal{F}_{\text{MMI}}$ criterion can also be used to train a GMM-HMM system [107, 142, 160]. Unlike the $\mathcal{F}_{\text{ML}}$ criterion of (2.67), which aims to train the joint observation sequence and hypothesis likelihood, the $\mathcal{F}_{\text{MMI}}$ criterion instead aims to train the hypothesis posteriors. Using (2.13), the $\mathcal{F}_{\text{MMI}}$ criterion can be expressed as

$$\mathcal{F}_{\text{MMI}}\left(\boldsymbol{\Phi}\right) = \log p\left(\boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T} \middle| \boldsymbol{\Phi}\right) - \log \sum_{\boldsymbol{\omega}} p\left(\boldsymbol{\omega}, \mathbf{O}_{1:T} | \boldsymbol{\Phi}\right). \tag{2.82}$$

By comparing $\mathcal{F}_{\text{MMI}}$ with $\mathcal{F}_{\text{ML}}$ in (2.67), it can be seen than maximising $\mathcal{F}_{\text{MMI}}$ not only maximises the likelihood of the manual transcriptions, but also simultaneously suppresses the likelihoods of competing hypotheses. A GMM-HMM can be trained toward the $\mathcal{F}_{\text{MMI}}$ criterion using the extended Baum-Welch algorithm [52, 107, 142].

The $\mathcal{F}_{\text{MMI}}$ criterion aims to maximise the conditional likelihood of the manual transcriptions. This aims for the model to exhibit a good hypothesis, or sentence-level, classification error rate. However, ASR systems are often assessed based on their WER performance. It may therefore be preferable to find the model parameters that directly minimise the WER. However, this must take into account the decoding process, and the Levenstein distance used to compute the WER is not differentiable. One possible solution is to instead minimise the expected risk, known as the Minimum Bayes' Risk (MBR) criterion [77],

$$\mathcal{F}_{\text{MBR}}\left(\boldsymbol{\Phi}\right) = \sum_{\boldsymbol{\omega}} \mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\omega}^{\text{ref}}\right) P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right). \tag{2.83}$$

If the risk, $\mathcal{L}$, is a word-level Levenstein distance of (2.7), then the criterion minimises the expected WER. It is also possible to measure the risk over other acoustic units. This leads to a variety of possible criteria, such as minimum word error for the word level, minimum phone error for the phone level [117], and state-level MBR (sMBR) for the per-frame state cluster level [48, 115]. Work in [132] suggests that using a word-level risk may result in a better WER performance, because this risk is well matched with the WER metric. However, work in [48, 117] suggests the opposite trend, where using a risk at a finer acoustic unit, such as phones or state clusters, may result in a better WER performance.

2.4 Neural network training

This section discusses how NN parameters can be optimised toward the training criteria discussed in Section 2.3.

2.4.1 Gradient descent

NN training is often a non-convex problem [25], and training methods seek to find a set of parameters that is locally optimal. Gradient descent is one possible method of finding a local optimum of the training criterion. Gradient descent is an iterative algorithm, whereby at each iteration, the parameters are updated as

$$\boldsymbol{\Phi}^{(v)} = \boldsymbol{\Phi}^{(v-1)} - \eta_v \boldsymbol{\nabla}_{\boldsymbol{\Phi}}\, \mathcal{F}\left(\boldsymbol{\Phi}^{(v-1)}\right)\Big|_{\mathcal{D}}, \tag{2.84}$$

where v is the training iteration index, η_v is the learning rate, and $\mathcal{D}$ is the training data. Each gradient descent update in (2.84) requires the computation of the gradient, $\boldsymbol{\nabla}_{\boldsymbol{\Phi}}\mathcal{F}(\boldsymbol{\Phi})$, accumulated over all samples within the training data, $\mathcal{D}$. This can be computationally expensive when the training dataset is large. The Stochastic Gradient Descent (SGD) training method reduces this computational cost by only computing the gradient over a random subset of the training data samples at each iteration,

$$\boldsymbol{\Phi}^{(v)} = \boldsymbol{\Phi}^{(v-1)} - \eta_v \boldsymbol{\nabla}_{\boldsymbol{\Phi}} \mathcal{F}\left(\boldsymbol{\Phi}^{(v-1)}\right)\Big|_{\widetilde{\mathcal{D}}^{(v)}}, \tag{2.85}$$

where $\widetilde{\mathcal{D}}^{(v)}$ is a random subset of the training data samples, referred to as a mini-batch. The mini-batch gradient, $\boldsymbol{\nabla}_{\boldsymbol{\Phi}}\mathcal{F}|_{\widetilde{\mathcal{D}}^{(v)}}$, can be viewed as a Monte Carlo approximation to the gradient over the whole training dataset, $\boldsymbol{\nabla}_{\boldsymbol{\Phi}}\mathcal{F}|_{\mathcal{D}}$. Although the mini-batch gradient is less computationally expensive to compute, this Monte Carlo approximation may introduce some variance into the gradient estimate.

Often, the mini-batches are sampled without replacement, as this can be implemented simply by randomly shuffling the training data and dividing the shuffled data into the mini-batches. The bias of the mini-batch gradient estimates can be minimised by shuffling the data appropriately and having sufficiently large mini-batch sizes. This is to ensure that each mini-batch contains data from a variety of utterances, speakers, genders, and environmental conditions.

2.4.2 Back-propagation

The gradient-based training methods discussed in Section 2.4.1 require the computation of the derivatives of the training criterion with respect to each of the model parameters, referred to as the gradient. The error back-propagation algorithm [123] can be used to compute these derivatives efficiently. Consider the training of a feed-forward DNN with K hidden layers, whose forward pass is described by (2.35) to (2.38). This forward pass involves passing the hidden layer activations, $\mathbf{h}_t^{(k)}$, from one layer to the next. Analogously, the gradient can be computed by passing "error signals" backward from each layer to the previous through the DNN.

The back-propagation algorithm begins by first computing the derivative of the training criterion with respect to the output layer pre-softmax activations, $\frac{\partial \mathcal{F}}{\partial \mathbf{z}_t^{(K+1)}}$, for each frame. The form of this derivative will depend on the choice of training criterion. Using the chain

rule, the error signals passed from each layer to the previous layer are

$$\frac{\partial \mathcal{F}}{\partial \mathbf{h}_t^{(k)}} = \boldsymbol{\nabla}_{\mathbf{h}_t^{(k)}} \mathbf{z}_t^{(k+1)} \frac{\partial \mathcal{F}}{\partial \mathbf{z}_t^{(k+1)}} \tag{2.86}$$

$$= \mathbf{W}^{(k+1)^{\mathrm{T}}} \frac{\partial \mathcal{F}}{\partial \mathbf{z}_t^{(k+1)}}. \tag{2.87}$$

Starting from the output layer, these error signals are passed back from layer to layer till the input layer. For each of the hidden layers, the derivative of the training criterion with respected to the pre-nonlinearity activations can be computed as

$$\frac{\partial \mathcal{F}}{\partial \mathbf{z}_t^{(k)}} = \boldsymbol{\nabla}_{\mathbf{z}_t^{(k)}} \mathbf{g}^{(k)} \left(\mathbf{z}_t^{(k)} \right) \frac{\partial \mathcal{F}}{\partial \mathbf{h}_t^{(k)}}. \tag{2.88}$$

The form of the derivative, $\boldsymbol{\nabla}_{\mathbf{z}_t^{(k)}} \mathbf{g}^{(k)} \left(\mathbf{z}_t^{(k)} \right)$, depends on the activation function used. The gradients for the NN parameters at each layer can be computed from the pre-nonlinearity activation derivatives as

$$\left. \frac{\partial \mathcal{F}}{\partial \mathbf{W}^{(k)}} \right|_{\mathcal{D}_t} = \boldsymbol{\nabla}_{\mathbf{W}^{(k)}} \mathbf{z}_t^{(k)} \frac{\partial \mathcal{F}}{\partial \mathbf{z}_t^{(k)}} \tag{2.89}$$

$$= \mathbf{h}_t^{(k)^{\mathrm{T}}} \frac{\partial \mathcal{F}}{\partial \mathbf{z}_t^{(k)}} \tag{2.90}$$

$$\left. \frac{\partial \mathcal{F}}{\partial \mathbf{b}^{(k)}} \right|_{\mathcal{D}_t} = \boldsymbol{\nabla}_{\mathbf{b}^{(k)}} \mathbf{z}_t^{(k)} \frac{\partial \mathcal{F}}{\partial \mathbf{z}_t^{(k)}} \tag{2.91}$$

$$= \frac{\partial \mathcal{F}}{\partial \mathbf{z}_t^{(k)}}, \tag{2.92}$$

where $\mathcal{D}_t$ represents the tth frame of data. These gradients can then be accumulated across all training data samples within a mini-batch for SGD. Passing the error signals, $\frac{\partial \mathcal{F}}{\partial \mathbf{h}_t^{(k)}}$, from the output layer down the NN layers allows the parameter gradients to be computed efficiently. The back-propagation algorithm can be extended to also propagate error signals across time steps to train an RNN, using back-propagation through time [153].

2.4.3 Parameter initialisation

Iterative training methods, such as the gradient-based methods described in Section 2.4.1, require an initial set of model parameters to begin training from. It has been found that the choice of parameter initialisation has a significant impact on the ability of the iterative training method to converge to a good local optimum [68]. Having NN parameters that are

too large or small can result in gradients exploding or vanishing as they are back-propagated through the NN [50]. This effect can be exacerbated by the saturation of the nonlinear activation functions [50]. Furthermore, many local optima may exist across the error surface of the training criterion [25]. As such, it may be important to properly initialise the model parameters to avoid converging to poor local optima.

Random initialisation

The NN parameters can be randomly initialised, by sampling the NN weights from a probability density function. It is important to set the variance of this probability density function, such that the sampled weights have an appropriate dynamic range [50]. If the parameters are too closely centred around zero, then the magnitude of the error signals may decrease rapidly when being back-propagated through the NN, because of the repeated application of (2.87). This may lead to very small updates for parameters in the lower layers of the NN. Similarly, if the parameters are too large, then the magnitude of the error signals may increase rapidly when being back-propagated. If the parameters are initialised to be too large, the pre-nonlinearity activations, $\mathbf{z}_t^{(k)}$, may also tend to be large, thereby causing the nonlinear activation function to operate primarily within a region of saturation. Within a saturated region, the derivative of $\boldsymbol{\nabla}_{\mathbf{z}_t^{(k)}} \mathbf{g}^{(k)} \left(\mathbf{z}_t^{(k)} \right)$ may be very small. This may again lead to very small updates of the NN parameters. There is no guarantee that the gradient shrinking caused by the activation function saturation will cancel out the gradient growth from the large parameters. As such, it is important to initialise the NN parameters within an appropriate dynamic range. To achieve this, the work in [50] proposes to sample the NN parameters as

$$w_{ij}^{(k)} \sim \mathcal{U} \left(-\frac{\sqrt{6}}{\sqrt{\dim\left(\mathbf{h}^{(k-1)}\right) + \dim\left(\mathbf{z}^{(k)}\right)}}, \frac{\sqrt{6}}{\sqrt{\dim\left(\mathbf{h}^{(k-1)}\right) + \dim\left(\mathbf{z}^{(k)}\right)}} \right) \tag{2.93}$$

$$b_i^{k)} = 0, \tag{2.94}$$

where $\mathcal{U}(\cdot, \cdot)$ is a continuous uniform probability density function. Here, the dynamic range of the probability density function is controlled by the dimensions of the layers.

Generative, unsupervised pre-training

After the NN parameters have been randomly initialised, pre-training methods can be used to prevent convergence to poor local optima. These pre-training methods often train the NN in a layer-wise fashion, to allow the hidden layers of the NN to develop representations that gradually improve, going up the NN. Here, the NN is trained in an iterative fashion, where

at each iteration, an additional layer is appended unto to NN. The previously trained layers are kept fixed, and only the newly appended layer is trained. The NN can be pre-trained toward generative criteria in an unsupervised fashion or toward discriminative criteria in a supervised fashion.

One generative pre-training method is to treat the NN as a stacked Restricted Boltzmann Machine (RBM) [68]. Here, the NN is no longer treated as a disciminative classifier, but as an undirected graphical model, from which the NN input observations, $\mathbf{o}_t$, can be generated from. As each new layer is appended, the inputs to the layer, $\mathbf{h}^{(k-1)}$, are treated as observed variables, while the outputs of the layer, $\mathbf{h}^{(k)}$, are treated as unobserved variables. At the lowest layer, the observed variables correspond to the NN input observations, $\mathbf{h}_t^{(0)} = \mathbf{o}_t$. The unobserved variables at the outputs of each layer are treated as the observed variables at the inputs of the next layer. Graphical connections are only allowed between observed and unobserved variables within each layer. In the RBM, the joint likelihood of both the observed and unobserved variables is given by

$$p\left(\mathbf{h}^{(k-1)}, \mathbf{h}^{(k)} \middle| \mathbf{\Phi}^{(k)}\right) = \frac{1}{Z} \exp\left[-E\left(\mathbf{h}^{(k-1)}, \mathbf{h}^{(k)} \middle| \mathbf{\Phi}^{(k)}\right)\right], \tag{2.95}$$

where the normalisation term is

$$Z = \sum_{\mathbf{h}^{(k-1)}, \mathbf{h}^{(k)}} \exp\left[-E\left(\mathbf{h}^{(k-1)}, \mathbf{h}^{(k)} \middle| \mathbf{\Phi}^{(k)}\right)\right]. \tag{2.96}$$

The energy function[3] is defined as

$$E\left(\mathbf{h}^{(k-1)}, \mathbf{h}^{(k)} \middle| \mathbf{\Phi}^{(k)}\right) = -\mathbf{b}^{(k)\prime\mathrm{T}}\mathbf{h}^{(k-1)} - \mathbf{b}^{(k)\mathrm{T}}\mathbf{h}^{(k)} - \mathbf{h}^{(k)\mathrm{T}}\mathbf{W}^{(k)}\mathbf{h}^{(k-1)}, \tag{2.97}$$

where, the parameters of the kth RBM layer are $\mathbf{\Phi}^{(k)} = \left\{\mathbf{W}^{(k)}, \mathbf{b}^{(k)}, \mathbf{b}^{(k)\prime}\right\}$.

At the lowest layer, the observed variables correspond to the NN input observations, $\mathbf{h}_t^{(0)} = \mathbf{o}_t$. For a RBM with only a single layer. the observation likelihood can be computed as

$$p\left(\mathbf{o}_t \middle| \mathbf{\Phi}^{(1)}\right) = \sum_{\mathbf{h}^{(1)}} p\left(\mathbf{o}_t, \mathbf{h}^{(1)} \middle| \mathbf{\Phi}^{(1)}\right). \tag{2.98}$$

This RBM is trained by maximising the observation log-likelihood,

$$\mathcal{F}_{\text{RBM}}\left(\mathbf{\Phi}^{(1)}\right) = \sum_{t=1}^{T} \log p\left(\mathbf{o}_t \middle| \mathbf{\Phi}^{(1)}\right). \tag{2.99}$$

[3]Here a simple RBM configuration is presented, where it is assumed that both the observed and unobserved variables are have elements that can take binary values.

The parameters can be trained in a computationally efficient manner using the contrastive divergence algorithm [66]. Successive layers can be trained using the outputs of previous layers as observed variables, and maximising the likelihood of these observed variables.

The stacked-RBM generative pre-training method is unsupervised, as it does not require output targets in the training data. As such, it is possible to perform generative pre-training on unlabelled training data.

Discriminative, supervised pre-training

Pre-training the NN parameters in a generative fashion allows the hidden representations to learn to capture abstract information about the input observations. However, this may not be optimal for the ASR task, which relies on the computation of the state cluster posteriors at the NN outputs. Rather than pre-training each layer to maximise the likelihood of the observations using $\mathcal{F}_{\text{RBM}}$ in (2.99), it may instead be better to pre-train each layer to discriminate between the output state clusters [9], using supervised pre-training. This iteratively trains the NN in a layer-wise fashion toward a supervised criterion, which takes into account both the inputs and outputs of the NN. Starting from an NN with only a single hidden layer, a new hidden layer is appended to the NN at each iteration, and the new hidden layer is trained greedily toward the supervised criterion. A possible choice of supervised criterion is the cross-entropy criterion in (2.80). However, unlike generative pre-training, supervised pre-training requires labelled training data, which may be more expensive to obtain.

These layer-wise pre-training methods can be used to initialise the hidden layer parameters, to produce sensible hidden representations, with the intention of avoiding poor local optima. After pre-training, a randomly sampled output layer can be appended to the final hidden layer, and the NN is then trained as a whole.

2.4.4 Regularisation

A trained model may perform well on the data that has been used for training. However, it is often more important for the trained model to be able to perform well on unseen data. Regularisation methods can be used to reduce over-fitting to the training data and improve generalisation to unseen data.

L2 regularisation

One possible regularisation method is to explicitly add a regularisation term, $\mathcal{R}(\boldsymbol{\Phi})$, to the training criterion, $\mathcal{F}(\boldsymbol{\Phi})$,

$$\mathcal{F}'(\boldsymbol{\Phi}) = \mathcal{F}(\boldsymbol{\Phi}) + \mathcal{R}(\boldsymbol{\Phi}). \tag{2.100}$$

This regularisation term can be used to penalise the model complexity. Because of their large number of parameters, NN models often have large modelling capacities, and may be able to memorise and over-fit to the training data. This can be discouraged by imposing a penalty on the model complexity. One commonly used form of regularisation term is the square of the L2-norm of the parameters [10],

$$\mathcal{R}(\boldsymbol{\Phi}) = \varrho \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}, \tag{2.101}$$

where ϱ is the regularisation coefficient that is used to control the amount of regularisation. This form of regularisation encourages the NN parameters to take values near zero, which may in turn encourage the nonlinear activation functions of the NN to operate within their linear regions.

Applying regularisation of the form of (2.100) to the $\mathcal{F}_{\text{CE}}$ or $\mathcal{F}_{\text{MMI}}$ criteria in (2.80) and (2.81), can be interpreted as enforcing a prior distribution over the parameters. Finding the most probable, or Maximum a-Posteriori (MAP), set of model parameters can be expressed as

$$\begin{aligned} \boldsymbol{\Phi}^* &= \arg\max_{\boldsymbol{\Phi}} \{p(\boldsymbol{\Phi}|\mathcal{D})\} && (2.102)\\ &= \arg\max_{\boldsymbol{\Phi}} \{\log p(\mathcal{D}|\boldsymbol{\Phi}) + \log p(\boldsymbol{\Phi})\} && (2.103)\\ & && (2.104) \end{aligned}$$

where $p(\boldsymbol{\Phi})$ is a prior distribution over the parameters. If it is assumed that all frames in the training data are independent of each other, then

$$\boldsymbol{\Phi}^* = \arg\max_{\boldsymbol{\Phi}} \{\mathcal{F}_{\text{CE}}(\boldsymbol{\Phi})|_{\mathcal{D}} + \log p(\boldsymbol{\Phi})\}. \tag{2.105}$$

If it is instead assumed that all utterances are independent of each other, then

$$\boldsymbol{\Phi}^* = \arg\max_{\boldsymbol{\Phi}} \{\mathcal{F}_{\text{MMI}}(\boldsymbol{\Phi})|_{\mathcal{D}} + \log p(\boldsymbol{\Phi})\}. \tag{2.106}$$

The regularisation term in (2.100) can be interpreted as the prior term in (2.105) and (2.106), which for L2 regularisation, leads to a prior of

$$p_{\text{L2}}\left(\mathbf{\Phi}\right) = \frac{1}{Z}\exp\left[\varrho\mathbf{\Phi}^{\text{T}}\mathbf{\Phi}\right], \tag{2.107}$$

where Z is a normalisation term to ensure that the prior is a valid probability density function. L2 regularisation therefore enforces a Gaussian prior onto the parameters, where the variance is controlled by the regularisation coefficient, ϱ. Applying more regularisation by increasing ϱ reduces the prior variance and encourages the parameters to take values closer to zero.

Early stopping

The measured training criterion cost on both the training and unseen data often improve in the early epochs of training. However, at later epochs, the cost measured on unseen data can degrade, while that measured on the training data continues to improve. This is an indication that the model has begun to over-fit to the training data. Early stopping methods aim to stop training near where the cost measured on unseen data is at a minimum, and may result in a model that generalises better. This may prevent the model parameters from specialising too much toward the training data.

To perform early stopping, the full dataset is often split into a training set and a held-out validation set. Typically, around 90% of the data is allocated to the training set, while the remaining 10% is allocated to the validation set. These data subsets do not overlap. The model parameters are trained only on the training set, and the validation set is used to estimate the model performance on unseen data. The NewBob training scheduler [119] is one variant of early stopping. Here, the training criterion cost measured on the validation set is used to dynamically control the learning rate and decide when to stop training.

Dropout

Dropout [137] is another method that can be used for regularisation. When an NN over-fits to the training data, certain hidden units may become overly specialised to modelling particular aspects of the training data. Dropout aims to prevent such specialisation, by randomly deactivating the hidden units during training and setting them to zero. This is achieved by multiplying the hidden activations, $\mathbf{h}_t^{(k)}$, with a vector of binary random variables, $\mathbf{d}^{(k)}$, in an element-wise fashion, such that the pre-nonlinearity activations, $\mathbf{z}_t^{(k)}$, of (2.36) are now computed as

$$\mathbf{z}_t^{(k)} = \frac{1}{1-\pi}\mathbf{W}^{(k)}\mathbf{Diag}\left(\mathbf{d}^{(k)}\right)\mathbf{h}_t^{(k-1)} + \mathbf{b}^{(k)}. \tag{2.108}$$

Here, $\mathbf{d}^{(k)}$ is known as a Dropout mask, $\textbf{Diag}\left(\mathbf{d}^{(k)}\right)$ is a diagonal matrix whose elements are those of $\mathbf{d}^{(k)}$, and π is the Dropout rate, which determines the expected fraction of hidden units that are deactivated. The $\frac{1}{1-\pi}$ factor in (2.108) allows the dynamic range of $\mathbf{z}_t^{(k)}$ to remain constant, regardless of the Dropout rate. Each element of the Dropout mask is a binary random variable, $d_i^{(k)} \in \{0, 1\}$, and is sampled independently from a Bernoulli distribution,

$$P\left(d_i^{(k)}\middle|\pi\right) = \pi^{1-d_i^{(k)}}\left(1-\pi\right)^{d_i^{(k)}}. \tag{2.109}$$

The random deactivation of hidden units may prevent each hidden unit from specialising toward particular aspects of the training data. Deactivating the hidden units also reduces the effective hidden layer dimensions of the NN during training, thereby potentially reducing its modelling capacity. When performing recognition, Dropout is often not used.

2.5 Derivative computation of sequence discriminative criteria

The NN parameters can be trained using SGD, and the gradients can be obtained using the back-propagation algorithm, as has been described in Sections 2.4.1 and 2.4.2. Section 2.4.2 explains that the back-propagation algorithm requires the derivative of the training criterion with respect to the pre-softmax activations, $\frac{\partial \mathcal{F}}{\partial \mathbf{z}_t^{(K+1)}}$. For the frame-level cross-entropy criterion in (2.80), this derivative is

$$\frac{\partial \mathcal{F}_{\text{CE}}\left(\mathbf{\Phi}\right)}{\partial z_{st}^{(K+1)}} = \delta\left(s, s_t^{\text{ref}}\right) - P\left(s|\mathbf{o}_t, \mathbf{\Phi}\right). \tag{2.110}$$

This derivative can be back-propagated down the NN to obtain the gradient updates for NN parameters.

Similarly, the derivatives of the sequence discriminative criteria need to be obtained. The derivative for the $\mathcal{F}_{\text{MMI}}$ criterion of (2.81) is [86]

$$\frac{\partial \mathcal{F}_{\text{MMI}}\left(\mathbf{\Phi}\right)}{\partial z_{st}^{(K+1)}} = \kappa\left[P\left(s_t = s\middle|\boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \mathbf{\Phi}\right) - P\left(s_t = s|\mathbf{O}_{1:T}, \mathbf{\Phi}\right)\right], \tag{2.111}$$

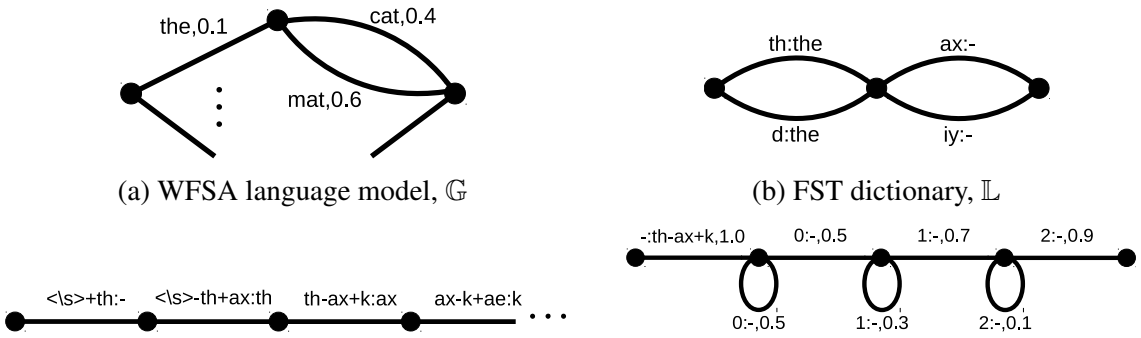
while that for the $\mathcal{F}_{\text{MBR}}$ criterion of (2.83) is [143]

$$\frac{\partial \mathcal{F}_{\text{MBR}}\left(\mathbf{\Phi}\right)}{\partial z_{st}^{(K+1)}} = \kappa P\left(s_t = s|\mathbf{O}_{1:T}, \mathbf{\Phi}\right)\sum_{\boldsymbol{\omega}}\mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\omega}^{\text{ref}}\right)\left[P\left(\boldsymbol{\omega}|s_t = s, \mathbf{O}_{1:T}, \mathbf{\Phi}\right) - P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \mathbf{\Phi}\right)\right], \tag{2.112}$$

where κ is the acoustic scaling factor.

The $\mathcal{F}_{\text{MMI}}$ derivative requires the computation of a "numerator" term, $P\left(s_t \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)$, and a "denominator" term, $P\left(s_t | \mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)$. It is possible to compute these terms using n-best lists of competing hypotheses. However, this method can become computationally impractical as the number of competing hypotheses grows. It is more efficient to represent the competing hypotheses in a lattice [108]. Similarly the terms in the derivative of the $\mathcal{F}_{\text{MBR}}$ criterion can also be computed over a lattice of competing hypotheses, as is described in [113].

2.5.1 Graphs and lattices



(a) WFSA language model, $\mathbb{G}$

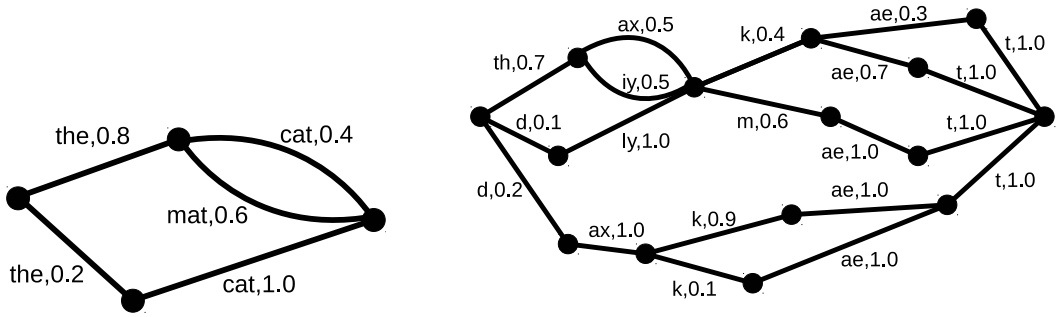
(b) FST dictionary, $\mathbb{L}$

(c) FST context-dependence, $\mathbb{C}$.

(d) WFST alignment model, $\mathbb{H}$.

Fig. 2.9 Weighted finite state acceptor and transducer graphs. These define the allowed transitions and the associated scores.

The language model, dictionary, context-dependence, and alignment model determine the word, phone, and state sequences that are allowed, and together, they define the set of possible hypotheses. These models can be compactly represented using graphs. The language model can be represented using a Weighted Finite State Acceptor (WFSA) graph, containing the allowed word sequences that can be accepted, with weights representing the language model probabilities. A word-level WFSA bigram language model, $\mathbb{G}$, is shown in Figure 2.9a. The dictionary, context-dependence, and alignment model can be represented by Weighted Finite State Transducer (WFST) graphs, mapping from context-independent sub-word units to words, context-dependent sub-word units to context-independent sub-word units, and states to context-dependent sub-word units respectively. Examples of WFSTs for the dictionary, $\mathbb{L}$, context-dependence, $\mathbb{C}$, and alignment model, $\mathbb{H}$, are shown in Figures 2.9b, 2.9c, and 2.9d respectively. The weights of the $\mathbb{H}$ graph represent the HMM transition probabilities, from the alignment model.



(a) Word-marked lattice.

(b) Phone-marked lattice.

Fig. 2.10 Lattice arcs marked with words and phones. These are utterance-specific. The node times are indicated by their horizontal positions.

These graphs are independent of the utterance and acoustic model, and do not contain any time information about when the transitions occur. In order to compute the sequence criteria derivatives, lattices are generated from these graphs, incorporating acoustic scores from the observation likelihoods of the acoustic model that is being trained and having lengths specific to each utterance [108]. These lattices also do not contain cycles. To generate the lattice of allowed hypotheses, the WFSA and WFST graphs can be composed together. Using the $\mathbb{G}$ graph alone can generate a lattice whose arcs are marked with words, shown in Figure 2.10a. This can be composed with the $\mathbb{L}$ graph to give $\mathbb{L} \circ \mathbb{G}$, which can generate a lattice whose arcs are marked with sub-word units, shown in Figure 2.10b. The composition can be continued all the way to form the $\mathbb{H} \circ \mathbb{C} \circ \mathbb{L} \circ \mathbb{G}$ recognition graph, to generate a lattice whose arcs are marked with states.

The numerator and denominator terms in the $\mathcal{F}_{\text{MMI}}$ derivative of (2.111) can be computed using forward-backward operations over a lattice, $\mathbb{A}$ [108]. The denominator term, $P\left(s_t|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)$, uses a lattice with all hypotheses that are allowed by the composite $\mathbb{H} \circ \mathbb{C} \circ \mathbb{L} \circ \mathbb{G}$ recognition graph, referred to as the denominator lattice. The numerator term, $P\left(s_t\middle|\boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)$, uses a lattice generated by constraining $\mathbb{G}$ to only accept the manual transcription word sequence, referred to as the numerator lattice. The denominator term can be computed using a forward-backward operation over the denominator lattice, as

$$P\left(s_t = s|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right) = \sum_{a_t \in \mathbb{A}: a_t \equiv s} \frac{\alpha\left[a_t\right] \beta\left[a_t\right]}{\sum\limits_{a_T \in \mathbb{A}} \alpha\left[a_T\right]}, \tag{2.113}$$

where a_t are the arcs at time t, which are here marked with state clusters. The forward, $\boldsymbol{\alpha}$, and backward, $\boldsymbol{\beta}$, probabilities represent

$$\frac{\alpha\left[a_t\right]}{Z} = p\left(\mathbf{O}_{1:t}, a_t|\boldsymbol{\Phi}\right) \tag{2.114}$$

$$\frac{\beta\left[a_t\right]}{Z} = p\left(\mathbf{O}_{t+1:T}|a_t, \boldsymbol{\Phi}\right), \tag{2.115}$$

where the normalisation factor is

$$Z = \sum_{a_T \in \mathbb{A}} \alpha\left[a_T\right]. \tag{2.116}$$

These can be computed recursively as

$$\alpha\left[a_t\right] = \sum_{a_{t-1} \in \mathbb{A}} \alpha\left[a_{t-1}\right] P\left(a_t|a_{t-1}\right) \mathcal{A}\left(\mathbf{o}_t, s \equiv a_t, \boldsymbol{\Phi}\right) \tag{2.117}$$

$$\beta\left[a_t\right] = \sum_{a_{t+1} \in \mathbb{A}} \beta\left[a_{t+1}\right] P\left(a_{t+1}|a_t\right) \mathcal{A}\left(\boldsymbol{o}_{t+1}, s \equiv a_{t+1}, \boldsymbol{\Phi}\right), \tag{2.118}$$

where $P\left(a_t|a_{t-1}\right)$ are the graph weights obtained from the language and alignment models.

The $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ probabilities need to be computed and stored for all arcs within the lattice. The number of arcs can be very large for lattices generated from the composite $\mathbb{H} \circ \mathbb{C} \circ \mathbb{L} \circ \mathbb{G}$ recognition graph. As such, it can be computationally expensive to compute the denominator derivative term over the whole lattice. Often, a simple language model, such as a unigram word-level model, is used during training, to allow deficiencies in the acoustic model to be more apparent [127]. The use of this simple word-level language model can help to reduce the number of arcs. However, in practice, the resulting lattice is often still computationally impractical to use.

One possible method of reducing this computational cost is to prune the denominator lattice, to only include the most likely hypotheses [108, 159]. Pruning the lattice leads to a reduction in the number of arcs, thereby reducing the number of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ probabilities that need to be computed. Using a pruned lattice to compute the derivative shall be referred to as the lattice-based method. However, pruning may result in a reduced diversity of hypotheses contained within the lattice. The quality of the resulting pruned lattice can be measured using the lattice word error rate [159], which computes the WER of the hypothesis contained within the lattice that is closest to the reference. This provides a lower bound on the WER that can be obtained by rescoring the lattice. Also, the computed derivative may be biased toward the most likely hypotheses. An unbiased estimate of the derivative can be obtained

by instead randomly sampling a finite number of lattice paths, and taking a Monte Carlo approximation to the derivative [132].

However, the pruning operation requires the acoustic scores, $\mathcal{A}\left(\mathbf{o}_t, s, \boldsymbol{\Phi}\right)$, to determine which hypotheses are more likely. This requires an existing acoustic model to compute. As such, a frame-level cross-entropy-trained model is often used as an initial model, to provide reasonable acoustic scores to prune the lattice for lattice-based sequence discriminative training [108]. Frame-level training is therefore required as a first step before lattice-based sequence discriminative training can be performed. This initial acoustic model is often also used as the parameter initialisation to begin sequence discriminative training from, as using a different parameter initialisation may result in a mismatch between the pruned lattice and the acoustic model. However, this frame-level parameter initialisation may lead to the model being biased toward the cross-entropy forced alignments. Furthermore, the limited variety of transition times contained within a pruned lattice may limit how far the sequence-trained system time alignment behaviour can diverge from that of the initial cross-entropy system.

2.5.2 Lattice-free training

An alternative method of reducing the computational cost of computing the derivatives of the sequence discriminative criteria is to simplify the graphs used to generate the lattices. The work in [116] proposes to use a 4-gram phone-level language model, $\mathbb{G}_{\text{phone}}$, instead of a unigram word-level language model, $\mathbb{G}$. This eliminates the need to compose the $\mathbb{G}$ and $\mathbb{L}$ graphs. thereby only requiring a composition of $\mathbb{H} \circ \mathbb{C} \circ \mathbb{G}_{\text{phone}}$ to generate the lattice. This leads to a large reduction in the number of arcs in the state-marked lattice. The number of arcs can be further reduced by simplifying $\mathbb{C}$ to use a biphone context-dependence instead of the usual triphones, and also by simplifying $\mathbb{H}$ to use a 2-state HMM topology shown in Figure 2.2b, instead of the usual 3-state HMM topology in Figure 2.2a. Another method proposed in [116] to reduce the computational cost of computing the derivatives is to use a frame shift of 30ms instead of the usual 10ms, thereby reducing the number of frames to about a third. When used with a 30ms frame shift, the 2-state HMM in Figure 2.2b allows each sub-word unit to be traversed in a minimum of 30ms, similarly to the minimum traversal time of the 3-state HMM in Figure 2.2a with a 10ms frame shift. It is shown in [116] that by using these simplifications, the number of lattice arcs can be reduced sufficiently to allow the derivatives to be computed over all hypotheses allowed by the $\mathbb{H} \circ \mathbb{C} \circ \mathbb{G}_{\text{phone}}$ composite graph, without needing to prune the generated lattices. The lattices that represent this hypothesis space that are specific to the length of each utterance and incorporate the acoustic scores can be generated on-the-fly during training. Therefore, pruned lattices for each utterance do not need to be pre-computed and stored. This is thus referred to as the lattice-free method.

Since pruning of the lattice is no longer required in the lattice-free method, it is possible to begin sequence discriminative training directly from a random parameter initialisation. This may remove any bias that the resulting trained model may have toward the cross-entropy forced alignments. Furthermore, since frame-level cross-entropy training is no longer required to provide an initial acoustic model, there is no need for the acoustic model to produce state cluster posteriors, $P\left(s|\mathbf{o}_t, \mathbf{\Phi}\right)$, at its output, as these posteriors are only required for the computation of the cross-entropy criterion and its derivative. In the lattice-based NN-HMM framework, the state cluster posteriors are converted to scaled observation likelihoods using (2.46). This requires the state cluster priors, $P\left(s\right)$, which also need to be estimated.

Without the need for cross-entropy training, it is possible to design the acoustic model to directly produce log-acoustic scores from a linear output layer [116], $\mathbf{z}_t^{(K+1)}$,

$$\log \mathcal{A}\left(\mathbf{o}_t, s, \mathbf{\Phi}\right) = z_{st}^{(K+1)}. \tag{2.119}$$

The hypothesis posteriors of (2.27) can then be computed as

$$P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \mathbf{\Phi}\right) = \frac{P^{\gamma}\left(\boldsymbol{\omega}\right) \sum\limits_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} \prod\limits_{t=1}^{T} P^{\gamma}\left(s_t|s_{t-1}\right) \mathcal{A}^{\kappa}\left(\mathbf{o}_t, s_t, \mathbf{\Phi}\right)}{\sum\limits_{\boldsymbol{\omega}'} P^{\gamma}\left(\boldsymbol{\omega}'\right) \sum\limits_{\mathbf{s}'_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}'}} \prod\limits_{t=1}^{T} P^{\gamma}\left(s'_t|s'_{t-1}\right) \mathcal{A}^{\kappa}\left(\mathbf{o}_t, s'_t, \mathbf{\Phi}\right)}. \tag{2.120}$$

However, the acoustic scores, $\mathcal{A}\left(\mathbf{o}_t, s, \mathbf{\Phi}\right)$, obtained in (2.119) are not valid probability density functions, as they are not normalised. These scores cannot even be treated as scaled likelihoods, as in (2.46), since they are not even normalised when multiplied by the observation prior, $p\left(\mathbf{o}_t\right)$. The hypothesis posterior in (2.120) is a valid probability distribution, as it is normalised and greater than or equal to zero. When computing these hypothesis posteriors, there is no need to estimate state cluster priors, $P\left(s\right)$.

The hypothesis posteriors in (2.120) can be interpreted as a product of experts, where the experts are the language, alignment, and acoustic models. This is similar to the Connectionist Temporal Classification (CTC) framework that is described in Section 2.6.1, which often takes a product of experts combination between the CTC acoustic model and a language model.

2.5.3 Lattices for recognition

Lattices are used not only for sequence discriminative training, but also when performing recognition. When performing recognition, the composite $\mathbb{H} \circ \mathbb{C} \circ \mathbb{L} \circ \mathbb{G}$ recognition graph

and the acoustic scores obtained from the acoustic model are used to generate a pruned lattice of competing hypotheses for each utterance. When performing recognition, the lattices need to contain information about the words in the hypotheses, and therefore both $\mathbb{L}$ and $\mathbb{G}$ need to be included in the composition of the recognition graph. Because of the inclusion of these graphs, the lattice needs to be pruned to reduce the number of arcs, and thereby limit the computational cost of performing recognition. Often, a weaker language model, such as a bigram or trigram word-level language model, is used to generate these pruned lattices for recognition. These weaker language models may exhibit language probabilities with higher entropies, which may allow the lattice to contain a wider diversity of hypotheses after pruning. The diverse lattice can then be rescored with stronger language models, such as a 4-gram or RNN language model, to improve the rank order of the hypotheses. Finally, the rescored lattice can be decoded to obtain the 1-best recognition hypothesis. Decoding can be performed using the Viterbi algorithm for MAP decoding, or either the recursive method proposed in [163] or the CN method to perform MBR decoding, as is discussed in Section 2.1.

2.6 Discriminative models

The HMM-based systems are generative models, which can be used to obtain hypothesis posteriors using (2.13). The resulting hypothesis posteriors will be correct if the joint distribution of $p(\boldsymbol{\omega}, \mathbf{O}_{1:T})$ is correct. However, the approximations used within HMM-based systems may limit the accuracy at which the systems can model speech data. An alternative approach is to directly model the hypothesis posteriors, $P(\boldsymbol{\omega}|\mathbf{O}_{1:T})$, using a discriminative model.

2.6.1 Connectionist temporal classification

One recently proposed form of discriminative model for ASR is Connectionist Temporal Classification (CTC) [55]. The hypothesis posteriors can be expressed as

$$P(\boldsymbol{\omega}|\mathbf{O}_{1:\mathrm{T}}, \boldsymbol{\Phi}) = \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P(\boldsymbol{\omega}|\mathbf{s}_{1:T}, \mathbf{O}_{1:T}) \prod_{t=1}^{T} P(s_t|\mathbf{s}_{1:t-1}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}) . \tag{2.121}$$

In CTC, graphemes are often used as the latent states, $\mathbf{s}_{1:T}$. If there are no homographemic words, then

$$P(\boldsymbol{\omega}|\mathbf{s}_{1:T}, \mathbf{O}_{1:T}) = \begin{cases} 1 & , \text{ if } \mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}} \\ 0 & , \text{ otherwise} \end{cases} . \tag{2.122}$$

CTC models are often trained with a grapheme-level manual transcription [55] and the dictionary is configured to allow all possible state sequences, such that $\sum_{\omega} P\left(\boldsymbol{\omega}|\mathbf{s}_{1:T}, \mathbf{O}_{1:T}\right) = 1$ for all $\mathbf{s}_{1:T}$. The hypothesis posterior of (2.121) is therefore a normalised probability distribution. Unlike the conditional independence assumptions of (2.24) and (2.25) made in an HMM-based system, the CTC model instead makes the assumption that the current state is conditionally independent of all other states, when given the observation sequence,

$$P\left(s_t|\mathbf{s}_{1:t-1}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}\right) \approx P\left(s_t|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right). \tag{2.123}$$

The per-frame state posteriors, $P\left(s_t|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)$, can be obtained as the outputs of a BLSTM. With this assumption, the CTC hypothesis posteriors of (2.121) can be expressed as [55]

$$P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right) = \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} \prod_{t=1}^{T} P\left(s_t|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right), \tag{2.124}$$

This represents a discriminative model, meant to directly compute the hypothesis posteriors, that can be used for classification.

The conditional independence between consecutive states in CTC may make it difficult to capture the dependencies between words or states over time, which are captured in the language and alignment models in the HMM-based systems. Several attempts have been made to incorporate a language model into the CTC system [56, 80]. Work in [56] has proposed to incorporate a separately-trained language model into CTC as,

$$P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right) = \frac{1}{Z\left(\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)} P^{\gamma}\left(\boldsymbol{\omega}\right) \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} \prod_{t=1}^{T} P^{\kappa}\left(s_t|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right), \tag{2.125}$$

which can be interpreted as a product of experts. Here, $Z\left(\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)$ ensures that the hypothesis posteriors are normalised. Similarly to the NN-HMM system, the dynamic ranges of the models can be adjusted for a better match using γ and κ.

Taking this product of experts extension even further, an alignment model can also be incorporated into CTC as

$$P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right) = \frac{1}{Z\left(\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)} P^{\gamma}\left(\boldsymbol{\omega}\right) \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} \prod_{t=1}^{T} P^{\gamma}\left(s_t|s_{t-1}\right) P^{\kappa}\left(s_t|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right). \tag{2.126}$$

The original conditional independence assumption between states in (2.123) can be obtained by using an alignment model for each sub-word unit with a topology shown in Figure 2.11, with the condition that all repeated states must be consumed by the self-loops. Setting the transition probabilities to those shown in Figure 2.11 results in an effective uniform transition

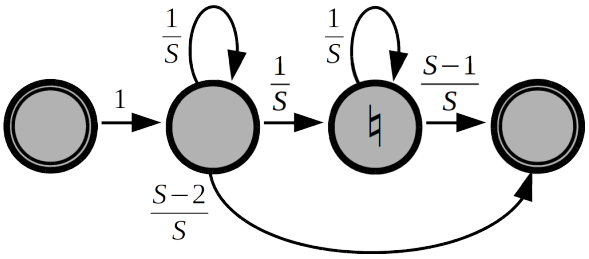


Fig. 2.11 Effective CTC alignment model topology. S is the total number of states and $\natural$ is a "blank" state. Setting the transition probabilities to the values shown results in uniform transition probability distributions.

probability between any two states,

$$P\left(s_t|s_{t-1}\right) = \frac{1}{S}, \tag{2.127}$$

where S is the total number of states. This transition probability is the same for all state pairs and implies the conditional independence assumption between states. The $\natural$ state in Figure 2.11 is referred to as a blank state [55]. Rather than having just a single state with a self-loop, the blank state allows for more model flexibility, by not requiring the NN to only output a single state classification throughout the duration of a sub-word unit. This can be especially beneficial when silences occur between words. The effective alignment model topology for CTC in Figure 2.11 is similar to that used for the lattice-free NN-HMM system in figure 2.2b. The difference is that the lattice-free alignment model only allows the first state to be traversed once. In CTC, the blank states often also have their acoustic scores tied across all sub-word units.

When using an acoustic model that takes into account the full observation context, $\mathbf{O}_{1:T}$, such as a BLSTM, the hypothesis posteriors for the hybrid NN-HMM system in (2.47) can be expressed as

$$P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right) = \frac{1}{Z\left(\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)} P^{\gamma}\left(\boldsymbol{\omega}\right) \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} \prod_{t=1}^{T} P^{\gamma}\left(s_t|s_{t-1}\right) \frac{P^{\kappa}\left(s_t|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)}{P^{\kappa}\left(s_t\right)}, \tag{2.128}$$

while those for the lattice-free NN-HMM system in (2.120) can be expressed as

$$P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right) = \frac{1}{Z\left(\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)} P^{\gamma}\left(\boldsymbol{\omega}\right) \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} \prod_{t=1}^{T} P^{\gamma}\left(s_t|s_{t-1}\right) \mathcal{A}^{\kappa}\left(\mathbf{O}_{1:T}, s_t, \boldsymbol{\Phi}\right). \tag{2.129}$$

Here again, $Z\left(\mathbf{O}_{1:T}, \boldsymbol{\Phi}\right)$, ensures that these hypothesis posteriors are normalised. Comparing (2.126), (2.128) and (2.129), it can be seen that all three forms of hypothesis posteriors can be

interpreted as products of language model $P(\boldsymbol{\omega})$, alignment model $P(s_t|s_{t-1})$, and acoustic model experts. The difference between the systems is the choice of acoustic model.

The CTC model can be trained in a discriminative manner, to maximise the conditional log-likelihood of the manual transcriptions [55], similarly to the $\mathcal{F}_{\text{MMI}}$ criterion in (2.81). For computational simplicity, the CTC model in the form of (2.124), without language or alignment models is used during training. This leads to a criterion of

$$\mathcal{F}_{\text{CTC}}(\boldsymbol{\Phi}) = \log \sum_{\mathbf{s}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}^{\text{ref}}}} \prod_{t=1}^{T} P(s_t|\mathbf{O}_{1:T}, \boldsymbol{\Phi}). \tag{2.130}$$

The derivative of this criterion is

$$\frac{\partial \mathcal{F}_{\text{CTC}}(\boldsymbol{\Phi})}{\partial z_{st}^{(K+1)}} = P\left(s_t = s \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}\right) - P(s_t = s|\mathbf{O}_{1:T}, \boldsymbol{\Phi}). \tag{2.131}$$

This is similar to the derivative of the $\mathcal{F}_{\text{MMI}}$ criterion in (2.111) for a hybrid NN-HMM system. In CTC, by allowing all possible state sequences and not using language or alignment models, there is no need for a normalising denominator in the hypothesis posteriors of (2.124). As such, the $P(s_t|\mathbf{O}_{1:T}, \boldsymbol{\Phi})$ term in the derivative can be taken directly as the NN acoustic model output, as opposed to the forward-backward operation that is required to compute the denominator term in the $\mathcal{F}_{\text{MMI}}$ derivative of (2.111). As such the sequence-level $\mathcal{F}_{\text{CTC}}$ criterion can be used to train a CTC model, without the need for pruned denominator lattices, and can therefore begin from a random parameter initialisation.

The cross-entropy criterion of (2.80) aims to maximise the conditional likelihood of the forced alignment state sequences. This may cause the trained model to be biased toward the forced alignments. The $\mathcal{F}_{\text{CTC}}$ criterion of (2.130) instead maximises the conditional likelihood of all possible alignments of the state sequences that can correspond to the manual transcriptions. This may reduce any bias that the trained model may have toward the target alignments.

2.6.2 Non-Markovian models

Both CTC and hybrid NN-HMM systems make strong conditional independence assumptions about the states and observations of (2.123), (2.24), and (2.25). These assumption may limit the ability to capture correlations in the data. Without making any conditional independence assumptions, the hypothesis posteriors can instead be decomposed into a product of word

posteriors,

$$P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T}\right) = \prod_{l=1}^{L} P\left(\omega_l|\boldsymbol{\omega}_{1:l-1}, \mathbf{O}_{1:T},\right). \tag{2.132}$$

However, this requires the relationship between word and observation sequences to be explicitly modelled. As is described in Section 2.2.2, this can be difficult, because of the potentially large difference between the lengths of word and observation sequences. Furthermore, the lengths of the word and observation sequences can vary between utterances.

One possible method to accommodate for the variable lengths of the sequences it to capture information about these sequences within fixed-dimensional representations,

$$P\left(\omega_l|\boldsymbol{\omega}_{1:l-1}, \mathbf{O}_{1:T}\right) \approx P\left(\omega_l\middle|\mathbf{h}_{l-1}^{\text{word}}, \mathbf{h}^{\text{obs}}, \boldsymbol{\Phi}^{\text{decode}}\right), \tag{2.133}$$

where the past word sequence can be represented as

$$\mathbf{h}_l^{\text{word}} = \mathbf{f}\left(\boldsymbol{\omega}_{1:l}, \boldsymbol{\Phi}^{\text{word}}\right), \tag{2.134}$$

and the full observation sequence can be represented as

$$\mathbf{h}^{\text{obs}} = \mathbf{f}\left(\mathbf{O}_{1:T}, \boldsymbol{\Phi}^{\text{obs}}\right). \tag{2.135}$$

This represents an encoder-decoder model topology. The separate $\boldsymbol{\Phi}^{\text{word}}$ and $\boldsymbol{\Phi}^{\text{obs}}$ encoders allow word and observation sequences of different lengths to be encoded into fixed-dimensional representations. These are then decoded using $\boldsymbol{\Phi}^{\text{decode}}$ to obtain the hypothesis posteriors. RNNs can be used for the encoders and decoder in the RNN-transducer framework [54]. It is also possible to use attention mechanisms to allow for a richer class of models [20].

These methods allow the full dependence between the word and observation sequences to be captured. However, the conditional likelihood of each word now depends on the full history of past words, similarly to an RNN language model, discussed in Section 2.2.1. As such, the Viterbi algorithm used for MAP decoding [146] and the methods for efficient MBR decoding [37, 100, 163] do not offer any computational savings when used with these systems.

2.7 Summary

This chapter has provided an overview of the HMM-based system for ASR. Methods for training and performing recognition have also been discussed. However, the HMM-based system makes strong conditional independence assumptions and is a generative model,

which can only be indirectly used to compute the hypothesis posteriors needed to perform recognition. Several discriminative models have been reviewed, that aim to overcome these limitations.

Chapter 3

Ensemble generation and combination

Chapter 2 has described how ASR can be approached using a single system. Often, significant performance gains can be obtained by combining together an ensemble of multiple systems. The gains that can be obtained through combination depend on the accuracy of the individual systems and the diversity between them [61]. This current chapter discusses methods for generating and combining an ensemble of diverse systems.

Section 3.1 relates ensemble methods to performing Bayesian inference of the hypothesis. Several methods are discussed that aim to sample models from various forms of probability density functions. However, these model sampling methods are often computationally expensive. As such, the simpler methods that aim to generate models with highly diverse behaviours, discussed in Section 3.2, are often used. Section 3.3 describes various forms of diversities that can be incorporated into an ensemble for ASR. Having generated an ensemble, the multiple models need to be combined together to perform recognition. Section 3.4 reviews several common methods of combining an ensemble. In Section 3.5, a novel method of obtaining feature-level diversity within an ensemble is proposed, by sampling random nonlinear recurrent feature projections. This can be used together with feature-level combination, for computational efficiency when performing recognition. Finally, it can often be useful to estimate the amount of diversity that an ensemble may capture. In Section 3.6, several methods are proposed to measure the ensemble diversity.

3.1 Bayesian neural network

The combination of an ensemble can be related to performing Bayesian inference. There are many design choices that need to be made when constructing a system for ASR. These include the choice of feature representations, model topologies and parameters, and sets of sub-word units and state clusters, to name a few. The choices made when constructing a

single system may not be the most appropriate for the task [97]. The Bayesian inference framework overcomes this, by marginalising over all possible model parameters, topologies, and design choices, such that the hypothesised word sequence can be inferred from the hypothesis posterior of [97]

$$P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\mathcal{D}\right)=\sum_{\mathcal{M}}\int P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\boldsymbol{\Phi},\mathcal{M}\right)p\left(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D}\right)P\left(\mathcal{M}|\mathcal{D}\right)d\boldsymbol{\Phi}, \tag{3.1}$$

where $\mathcal{D}$ is the training data and $\boldsymbol{\Phi}$ are the sets of acoustic model parameters. Here, the hypothesis posteriors from each system, $P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\boldsymbol{\Phi},\mathcal{M}\right)$, are written to explicitly show the dependence on $\mathcal{M}$, which represents the acoustic model topology, set of state clusters, set of sub-word units, feature representation, language model, alignment model, and other design aspects of the system architecture. The integral in (3.1) in a sense takes into account the uncertainty of which system design and parameters are most appropriate for the task.

For many problems of interest, it can be computationally intractable to marginalise over $\mathcal{M}$ and $\boldsymbol{\Phi}$. Work in [98] has investigated computing the model evidence, $P\left(\mathcal{D}|\mathcal{M}\right)$, which can then be used to estimate $P\left(\mathcal{M}|\mathcal{D}\right)$. However, these method tend to be computationally expensive. To reduce this computational cost, the hypothesis posteriors in (3.1) can be simplified by restricting a set of system design aspects, such as the model topology, to be fixed, leading to a constrained form of Bayesian inference,

$$P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\mathcal{M},\mathcal{D}\right)=\int P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\boldsymbol{\Phi},\mathcal{M}\right)p\left(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D}\right)d\boldsymbol{\Phi}. \tag{3.2}$$

However, this constraint may result in the inferred hypothesis being biased. Furthermore, forcing a common design upon all systems may limit how differently the systems can behave from each other, and therefore limit the diversity of behaviours that can be captured.

Bayesian inference can be performed in a computationally tractable manner by taking a Monte Carlo approximation. This approximates the hypothesis posteriors in (3.1) as

$$P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\mathcal{D}\right)\approx\sum_{m=1}^{M}\lambda_m P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\boldsymbol{\Phi}^m,\mathcal{M}^m\right), \tag{3.3}$$

where λ_m are the interpolation weights, satisfying $0\leq\lambda_m\leq 1$ and $\sum_m\lambda_m=1$. This is a combination of an ensemble of M systems, where m here is used to index the members of the ensemble. If the systems in the ensemble are sampled from the true posteriors, $\mathcal{M}^m\sim P\left(\mathcal{M}|\mathcal{D}\right)$ and $\boldsymbol{\Phi}^m\sim p\left(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D}\right)$, then an unbiased estimate of $P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\mathcal{D}\right)$ can be obtained by using equal interpolation weights, $\lambda_m=\frac{1}{M}$. However, it is often difficult to sample from $P\left(\mathcal{M}|\mathcal{D}\right)$. One possible method for sampling sets of model parameters from

$p(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D})$ is Markov chain Monte Carlo, discussed in Section 3.1.4. Alternatively, models can be sampled from approximate posteriors, $q(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D})$ and $Q(\mathcal{M}|\mathcal{D})$. Laplace's method, variational inference, and Monte Carlo Dropout aim to sample sets of model parameters from an approximate $q(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D})$, and are discussed later in this section. When sampling from an approximate posterior, an unbiased estimate of $P(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\mathcal{D})$ can be obtained by using importance sampling interpolation weights [99],

$$\lambda_m = \frac{p(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D})\,P(\mathcal{M}|\mathcal{D})}{q(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D})\,Q(\mathcal{M}|\mathcal{D})}. \tag{3.4}$$

However, it is often difficult to compute one or more of the distributions in (3.4). Alternatively, it is also possible to estimate the interpolation weights by other means, and accept the resulting bias in the inferred hypothesis. The interpolation weights can be explicitly optimised toward a training criterion [166], or they can be set based on a measure of confidence of each system [38]. However, to avoid complications that may arise when using these methods, simple equal interpolation weights can also be used.

This section discusses various methods for generating an ensemble by sampling sets of model parameters from a posterior. Since these methods only sample sets of model parameters, $\boldsymbol{\Phi}$, and not $\mathcal{M}$, the resulting combined ensemble can only be interpreted as an approximation to (3.2), and not (3.1). This may limit the diversity of the ensemble. Furthermore, these methods are often computationally expensive to use. They are therefore not used in this thesis. These methods are briefly reviewed here to demonstrate that it is possible to generate an ensemble that approximates Bayesian inference, through sampling.

3.1.1 Laplace's method

One simple approximate form of the model parameter posterior, $q(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D})$, is to use a Gaussian distribution centred around a local MAP model parameter set, known as Laplace's method. This is related to taking a second-order Taylor approximation of the log-posterior of the parameters around a local maximum of the posterior, $\boldsymbol{\Phi}^*$,

$$\begin{aligned}\log p(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D}) \approx &\log p(\boldsymbol{\Phi}^*|\mathcal{M},\mathcal{D}) + (\boldsymbol{\Phi}-\boldsymbol{\Phi}^*)^{\mathrm{T}}\,\boldsymbol{\nabla}_{\boldsymbol{\Phi}}\log p(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D})|_{\boldsymbol{\Phi}^*} \\ &+ \frac{1}{2}(\boldsymbol{\Phi}-\boldsymbol{\Phi}^*)^{\mathrm{T}}\,\mathbf{H}|_{\boldsymbol{\Phi}^*}(\boldsymbol{\Phi}-\boldsymbol{\Phi}^*),\end{aligned} \tag{3.5}$$

where $\mathbf{H}|_{\boldsymbol{\Phi}^*}$ is the local Hessian of the MAP model, whose elements are

$$h_{ij}|_{\boldsymbol{\Phi}^*} = \frac{\partial^2}{\partial\phi_i\partial\phi_j}\log p\left(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D}\right)\Big|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}^*} \tag{3.6}$$

$$= \frac{\partial^2}{\partial\phi_i\partial\phi_j}\log\left[p\left(\mathcal{D}|\boldsymbol{\Phi},\mathcal{M}\right)p\left(\boldsymbol{\Phi}|\mathcal{M}\right)\right]\Big|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}^*}. \tag{3.7}$$

Here, $\boldsymbol{\Phi}$ is a vector, whose elements, ϕ_i, represent each of the model parameters, which for an NN are the matrix weights and biases. The local MAP model parameters, $\boldsymbol{\Phi}^*$, can be found by optimising the regularised cross-entropy or sequence-level $\mathcal{F}_{\text{MMI}}$ criteria, as is described in Section 2.4.4. Since $\boldsymbol{\Phi}^*$ is a local maximum of $\log p\left(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D}\right)$, the first order term in (3.5) goes to zero. Taking the exponential of the Taylor approximation of (3.5) yields an approximate model parameter posterior that is a Gaussian centred at $\boldsymbol{\Phi}^*$,

$$q\left(\boldsymbol{\Phi}\Big|\boldsymbol{\Phi}^*,-\mathbf{H}^{-1},\mathcal{M},\mathcal{D}\right) = \frac{1}{Z}\exp\left[\frac{1}{2}\left(\boldsymbol{\Phi}-\boldsymbol{\Phi}^*\right)^{\mathsf{T}}\mathbf{H}\left(\boldsymbol{\Phi}-\boldsymbol{\Phi}^*\right)\right] \tag{3.8}$$

$$= \mathcal{N}\left(\boldsymbol{\Phi};\boldsymbol{\Phi}^*,-\mathbf{H}^{-1}\right), \tag{3.9}$$

where Z ensures that the posterior is normalised. An ensemble can be generated by sampling sets of model parameters from this approximate posterior,

$$\boldsymbol{\Phi}^m \sim q\left(\boldsymbol{\Phi}\Big|\boldsymbol{\Phi}^*,-\mathbf{H}^{-1},\mathcal{M},\mathcal{D}\right). \tag{3.10}$$

These model parameter samples can then be used to compute a Monte Carlo approximation to the constrained Bayesian inference of (3.2),

$$P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\mathcal{M},\mathcal{D}\right) \approx \sum_{m=1}^{M}\lambda_m P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\mathcal{M},\boldsymbol{\Phi}^m\right). \tag{3.11}$$

Laplace's method requires the computation of both $\boldsymbol{\Phi}^*$ and $\mathbf{H}^{-1}$. Storing the inverse Hessian requires memory space on the order of $\mathcal{O}\left(\dim^2\left(\boldsymbol{\Phi}\right)\right)$, and a naive implementation of computing the inverse Hessian has a complexity of $\mathcal{O}\left(\dim^3\left(\boldsymbol{\Phi}\right)\right)$. It may also be difficult to find an exact local maximum, $\boldsymbol{\Phi}^*$, when using an iterative training algorithm. Regularisation techniques, such as early stopping, can prevent a local maximum from being reached. Furthermore, it may be common for gradient descent based training methods to converge toward saddle points [30]. Using a non-maximal estimate of $\boldsymbol{\Phi}^*$ may lead to a first-order Taylor series term in (3.5) that is non-zero. Furthermore, an estimate of $\boldsymbol{\Phi}^*$ that is not exactly at a local maximum may have a local Hessian that is not negative definite. The Gaussian covariance matrix will then be singular. Another drawback of Laplace's method is that the

approximate model parameter posterior only has a single mode centred around one local maximum. This may limit the diversity of the models that are marginalised across.

3.1.2 Variational inference

Laplace's method approximates the model parameter posterior as a Gaussian at a local maximum of the $p(\boldsymbol{\Phi}|\mathcal{M}, \mathcal{D})$. It is also possible to use other forms of approximate model parameter posteriors. Variational inference is a general Bayesian method that can be used with many forms of approximate model parameter posteriors, $q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D})$, parameterised by $\boldsymbol{\Upsilon}$. In variational inference, the task of training $\boldsymbol{\Phi}$ is replaced with the task of training $\boldsymbol{\Upsilon}$, which defines a probability density function over $\boldsymbol{\Phi}$. An ensemble can again be generated by sampling sets of model parameters from $q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D})$ and combining them as (3.11).

In the variational inference framework [10], the log-likelihood of the data, referred to as the model evidence, is expressed as

$$\log p(\mathcal{D}|\mathcal{M}) = \mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon},\mathcal{M},\mathcal{D})}\left\{\log \frac{p(\mathcal{D}, \boldsymbol{\Phi}|\mathcal{M})}{q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}\mathcal{D})}\right\} + \mathrm{KL}\left\{q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D}) \| p(\boldsymbol{\Phi}|\mathcal{M}, \mathcal{D})\right\}. \tag{3.12}$$

This is similar to the EM algorithm described in Section 2.3.1. Here, the model parameters, $\boldsymbol{\Phi}$, are the latent variables, $\mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon},\mathcal{M},\mathcal{D})}\left\{\log \frac{p(\mathcal{D},\boldsymbol{\Phi}|\mathcal{M})}{q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon},\mathcal{M}\mathcal{D})}\right\}$ is the evidence lower bound, and $\mathrm{KL}\left\{q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D}) \| p(\boldsymbol{\Phi}|\mathcal{M}, \mathcal{D})\right\}$ is the KL-divergence between the approximate and true model parameter posteriors. Since the KL-divergence is non-negative, the evidence lower bound forms a lower bound to the model evidence.

Variational inference reduces the problem of Bayesian inference to training the approximate posterior parameters, $\boldsymbol{\Upsilon}$. These can be trained using the expectation step of the EM algorithm. That is, to minimise $\mathrm{KL}\left\{q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D}) \| p(\boldsymbol{\Phi}|\mathcal{M}, \mathcal{D})\right\}$. This seeks to find the approximate posterior that has the minimal KL-divergence distance from the true model parameter posterior. However, for many forms of $q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D})$ probability density functions, this KL-divergence can be difficult to optimise directly. Instead, since the KL-divergence is non-negative and the model evidence is independent of $\boldsymbol{\Upsilon}$, maximising the evidence lower bound, $\mathbb{E}_{\boldsymbol{\Phi}\sim q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon},\mathcal{M},\mathcal{D})}\left\{\log \frac{p(\mathcal{D},\boldsymbol{\Phi}|\mathcal{M})}{q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon},\mathcal{M}\mathcal{D})}\right\}$, will indirectly minimise the KL-divergence.

By using appropriate forms for the approximate model parameter posterior, $q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D})$, it can be easier to maximise the evidence lower bound than to minimise the KL-divergence. One simple form that can be used for $q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D})$ is a Gaussian with a diagonal covariance matrix,

$$q(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D}) = \mathcal{N}(\boldsymbol{\Phi}; \boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{3.13}$$

Here, $\boldsymbol{\Upsilon} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and diagonal covariance matrix. Using this form of approximate model parameter posterior, the evidence lower bound can be maximised using gradient descent, with update equations for the mean and covariance being similar to the standard gradient update in cross-entropy training [11]. This method is known as Bayes by backprop.

The flexibility of $q\left(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D}\right)$ is restricted by the chosen form of the probability density function. As such, a limited flexibility of $q\left(\boldsymbol{\Phi}|\boldsymbol{\Upsilon}, \mathcal{M}, \mathcal{D}\right)$ may in turn limit the ability to fully capture all aspects of the true model parameter posterior.

3.1.3 Monte Carlo Dropout

Dropout, as is described in Section 2.4.4, was originally introduced as a regularisation technique [137]. In its standard implementation, hidden nodes are randomly set to zero with some probability during training, using (2.108). When performing recognition, the standard Dropout method does not deactivate any hidden nodes, by setting the Dropout rate to $\pi = 0$.

Dropout can also be used to generate an ensemble and perform approximate Bayesian inference [137]. This is accomplished by using Dropout when performing recognition also, with $\pi > 0$. Let $\breve{\boldsymbol{\Phi}} = \left\{\mathbf{W}^{(k)}, \mathbf{b}^{(k)} \quad \forall k\right\}$ represent the set of model parameters without applying Dropout. Data can be fed through the NN multiple times, each time with a different Dropout mask sample, $\mathbf{d}^{(k)m}$. Each different Dropout mask sample can be used to form a separate member of the ensemble, whose effective parameters are $\boldsymbol{\Phi}^m = \left\{\mathbf{W}^{(k)m}, \mathbf{b}^{(k)m} \quad \forall k\right\}$, where

$$\mathbf{W}^{(k)m} = \frac{1}{1-\pi}\mathbf{W}^{(k)}\mathbf{Diag}\left(\mathbf{d}^{(k)m}\right) \tag{3.14}$$

$$\mathbf{b}^{(k)m} = \mathbf{b}^{(k)}. \tag{3.15}$$

The ensemble can then be combined using (3.11) [137]. Only a single model needs to be trained to generate the ensemble. The Bernoulli distribution from which $d_i^{(k)m}$ are sampled from in (2.109) defines the approximate model parameter posterior, $q\left(\boldsymbol{\Phi}\middle|\pi, \breve{\boldsymbol{\Phi}}, \mathcal{M}, \mathcal{D}\right)$ [42]. The probability density function from which each NN weight matrix row, $\mathbf{w}_j^{(k)m}$, is sampled from is

$$q\left(\mathbf{w}_j^{(k)m}\middle|\pi, \breve{\boldsymbol{\Phi}}, \mathcal{M}, \mathcal{D}\right) = \begin{cases} 1-\pi & , \text{ if } \mathbf{w}_j^{(k)m} = \mathbf{w}_j^{(k)} \\ \pi & , \text{ if } \mathbf{w}_j^{(k)m} = \mathbf{0} \\ 0 & , \text{ otherwise} \end{cases}. \tag{3.16}$$

This approximate model parameter posterior is non-zero only for a limited set of possible $\mathbf{\Phi}^m$ values. As such, sampling from this distribution may limit the diversity that the ensemble can capture.

3.1.4 Markov chain Monte Carlo

Laplace's method, Monte Carlo Dropout, and variational inference all sample the members of the ensemble from approximate posteriors, $q\left(\mathbf{\Phi}|\mathcal{M},\mathcal{D}\right)$. These methods also often impose constraints on the forms that $q\left(\mathbf{\Phi}|\mathcal{M},\mathcal{D}\right)$ can take. These constrained, approximate model parameter posteriors may not be able to fully represent the true model parameter posterior.

Rather than sampling the model parameters from an approximate posterior, Markov chain Monte Carlo methods instead aim to sample model parameters from the true model parameter posterior, $p\left(\mathbf{\Phi}|\mathcal{M},\mathcal{D}\right)$, using a Markov chain [35, 106, 151]. The standard Metropolis-Hastings method [63] for Markov chain Monte Carlo sampling is an iterative process, where the mth set of model parameters are sampled from a proposal density function,

$$\mathbf{\Phi}^m \sim p\left(\mathbf{\Phi}^m \middle| \mathbf{\Phi}^{m-1}\right), \tag{3.17}$$

and accepted with probability

$$\text{acceptance probability} = \min\left\{1, \frac{p\left(\mathbf{\Phi}^m|\mathcal{M},\mathcal{D}\right)}{p\left(\mathbf{\Phi}^{m-1}|\mathcal{M},\mathcal{D}\right)} \frac{p\left(\mathbf{\Phi}^{m-1}|\mathbf{\Phi}^m\right)}{p\left(\mathbf{\Phi}^m|\mathbf{\Phi}^{m-1}\right)}\right\}. \tag{3.18}$$

As long as the samples are accepted according to (3.18) and $p\left(\mathbf{\Phi}^m|\mathbf{\Phi}^{m-1}\right)$ covers the support of $p\left(\mathbf{\Phi}|\mathcal{M},\mathcal{D}\right)$, then the collection of $\mathbf{\Phi}^m$ will converge towards $p\left(\mathbf{\Phi}|\mathcal{M},\mathcal{D}\right)$ as $m \to \infty$ [63].

However, one drawback of Markov chain Monte Carlo sampling is that consecutive samples may not be independent of each other. This correlation between consecutive samples can be diminished by thinning the Markov chain. The correlation between consecutive samples and rate of exploration of the support of $\mathbf{\Phi}$ are strongly determined by the choice of proposal density function, $p\left(\mathbf{\Phi}^m|\mathbf{\Phi}^{m-1}\right)$. A simple choice is to use a Gaussian that is centred around the previous sample,

$$p\left(\mathbf{\Phi}^m \middle| \mathbf{\Phi}^{m-1}\right) = \mathcal{N}\left(\mathbf{\Phi}^m; \mathbf{\Phi}^{m-1}, \mathbf{\Sigma}\right), \tag{3.19}$$

where $\mathbf{\Sigma}$ is the covariance matrix that needs to be chosen. However, using this distribution can lead to slow random walk exploration when the $\mathbf{\Phi}$ is high dimensional.

Stochastic Gradient Langevin Dynamics (SGLD) [151] uses stochastic gradient information to improve the parameter space exploration. The proposal density function for SGLD is a Gaussian with a mean that is biased in the gradient direction of the previous parameter sample, computed over a mini-batch of data, $\widetilde{\mathcal{D}}^{(m)}$,

$$p\left(\boldsymbol{\Phi}^m\middle|\boldsymbol{\Phi}^{m-1}\right) = \mathcal{N}\left(\boldsymbol{\Phi}^m; \boldsymbol{\Phi}^{m-1} + \eta_m \, \boldsymbol{\nabla}_{\boldsymbol{\Phi}} \log p\left(\boldsymbol{\Phi}\middle|\mathcal{M}, \widetilde{\mathcal{D}}^{(m)}\right)\Big|_{\boldsymbol{\Phi}^{m-1}}, 2\eta_m \mathbf{I}\right), \tag{3.20}$$

where $\mathbf{I}$ is the identity matrix. This leads to a parameter update rule of,

$$\boldsymbol{\Phi}^m = \boldsymbol{\Phi}^{m-1} + \eta_m \, \boldsymbol{\nabla}_{\boldsymbol{\Phi}} \log p\left(\boldsymbol{\Phi}\middle|\mathcal{M}, \widetilde{\mathcal{D}}^{(m)}\right)\Big|_{\boldsymbol{\Phi}^{m-1}} + \boldsymbol{\vartheta}^m, \tag{3.21}$$

where η_m are the step sizes, and

$$\boldsymbol{\vartheta}^m \sim \mathcal{N}\left(\boldsymbol{\vartheta}; \mathbf{0}, 2\eta_m \mathbf{I}\right). \tag{3.22}$$

As is discussed in Section 2.4.4,

$$\boldsymbol{\nabla}_{\boldsymbol{\Phi}} \log p\left(\boldsymbol{\Phi}\middle|\mathcal{M}, \widetilde{\mathcal{D}}^{(m)}\right) = \boldsymbol{\nabla}_{\boldsymbol{\Phi}} \left[\mathcal{F}_{\text{CE}}\left(\boldsymbol{\Phi}\right)|_{\widetilde{\mathcal{D}}^{(m)}} + \log p\left(\boldsymbol{\Phi}|\mathcal{M}\right)\right], \tag{3.23}$$

if it is assumed that all frames of data are independent of each other. Here, $p\left(\boldsymbol{\Phi}|\mathcal{M}\right)$ is a prior over the model parameters. This leads to a Markov chain parameter update of

$$\boldsymbol{\Phi}^m = \boldsymbol{\Phi}^{m-1} + \eta_m \boldsymbol{\nabla}_{\boldsymbol{\Phi}} \left[\mathcal{F}_{\text{CE}}\left(\boldsymbol{\Phi}^{m-1}\right)\Big|_{\widetilde{\mathcal{D}}^{(m)}} + \log p\left(\boldsymbol{\Phi}^{m-1}\middle|\mathcal{M}\right)\right] + \boldsymbol{\vartheta}^m. \tag{3.24}$$

Comparing this update with (2.85), it can be seen that the SGLD update is simply the SGD update for the regularised cross-entropy criterion with added Gaussian noise. If it is assumed that all utterances are independent of each other, instead of all frames, then the update is

$$\boldsymbol{\Phi}^m = \boldsymbol{\Phi}^{m-1} + \eta_m \boldsymbol{\nabla}_{\boldsymbol{\Phi}} \left[\mathcal{F}_{\text{MMI}}\left(\boldsymbol{\Phi}^{m-1}\right)\Big|_{\widetilde{\mathcal{D}}^{(m)}} + \log p\left(\boldsymbol{\Phi}^{m-1}\middle|\mathcal{M}\right)\right] + \boldsymbol{\vartheta}^m. \tag{3.25}$$

Because of the stochasticity of the mini-batch, $\widetilde{\mathcal{D}}^{(m)}$, the proposal density function of (3.20) is not reversible, and it is therefore difficult to prove convergence using the acceptance probability of (3.18). If the step size is decayed sufficiently slowly to enable coverage of the parameter space, by ensuring that $\sum_m \eta_m = \infty$, then by using the Fokker-Planck equation, it can be proven that as $m \rightarrow \infty$ and $\eta_m \rightarrow 0$ at a rate such that $\sum_m \eta_m^2$ is finite [120], the parameter samples drawn using SGLD converge to a stationary probability density function of $p\left(\boldsymbol{\Phi}|\mathcal{M}, \mathcal{D}\right)$ [24]. This requires that the covariance of the added Gaussian noise, $2\eta_m\mathbf{I}$, is matched with the step size, η_m. However, when the step size is small, the consecutive

parameter samples will likely be highly correlated. The correlation can be reduced by thinning out intermediate samples.

SGLD is an approximation to the more general method of Hybrid/Hamiltonian Monte Carlo (HMC) [106]. HMC is a Markov chain Monte Carlo method inspired by Hamiltonian mechanics. In this method, a momentum latent variable is introduced, and the parameters are sampled from a joint proposal density function over the parameters and momentum. This involves simulating trajectories under conservation of energy. The use of momentum can improve the parameter space exploration. SGLD is a simplification of HMC, where the memory of the past momentum is forgotten after each simulation iteration. More advanced HMC-based methods, such as stochastic gradient Nosé-Hoover thermostats [35], improve the rate of exploration through the parameter space, at the expense of introducing additional latent variables that need to be sampled and stored. The parameter exploration rate can also be improved by preconditioning the gradient and Gaussian noise, allowing information about the local curvature of the parameter log-posterior to be taken into account [91].

The cost of generating each sample of model parameters in the Markov chain Monte Carlo method is similar to that of performing each iteration of SGD. Although this may not be too high, many sampling iterations may be needed to overcome the correlation between consecutive models and to obtain a final collection of diverse models.

3.2 General approaches to obtain diverse models

The methods discussed in Section 3.1 aim to generate an ensemble, by sampling sets of model parameters either from approximate model posteriors or the true model posterior. In so doing, the combination of the ensemble may be interpreted as a Monte Carlo approximation to performing Bayesian inference of the hypothesis. However, these methods only explore the space of model parameter, keeping the model topology and other aspects of the system design fixed. This may limit the diversity that can be captured within the ensemble. Many of these methods can also be computationally expensive to use. Furthermore, Laplace's method and Monte Carlo Dropout use forms of approximate model parameter posteriors that may further limit the ensemble diversity.

Rather than sampling models from posteriors of known forms, it is also common in practice to generate multiple systems with the aim of obtaining diverse behaviours between the systems. As is illustrated in [61], the combined performance of an ensemble depends both on the performance of each individual system and the diversity between the system behaviours. If all systems make the same errors in their hypotheses, then little can be gained from combination. If the systems make different errors in their hypotheses, then they

may be able to correct for each others' errors through combination [38]. The systems may make more different errors if they behave more differently. This section discusses several methods to encourage differences between the system behaviours, to obtain diversity within the ensemble. A simple method of obtaining a diverse ensemble is to use systems that have been constructed by independent teams [38]. Each team designs their own system, based on their own subjective biases and prior experience. Between the teams, these systems may span across a diverse range of system designs, and may therefore be highly diverse and complementary to each other.

3.2.1 Data selection

A simple method of generating multiple diverse systems is to train each system on a different subset of the training data. This method is known as bagging [13]. Since each system is trained only on a subset of the data, it is hoped that each system will specialise toward the data that it has seen. This may implicitly allow the systems to develop diverse behaviours. However, since each system is not trained on the full dataset, and may therefore not generalise well.

Boosting aims to increase the diversity, by explicitly training the systems to compensate for each others' weaknesses. One implementation of boosting, known as Adaboost [41], trains systems one after the other, with a weight applied to each training data sample, depending on how well the previously trained systems are able to model that sample. Therefore, each successive system tries to learn to specialise toward the training data samples that the previous systems are not able to adequately model. Each system is trained on the full dataset, and may therefore generalise better.

3.2.2 Feature subsampling

Another method of implicitly allowing the systems to develop diverse behaviours is to train each system on only a subset of the dimensions of the input features [16]. This trains each system to rely on different information in the input features. Unlike bagging, each system here is trained on all data samples in the full training dataset. This method can be viewed as a special case of Dropout, where a different static Dropout mask sample is applied to the input layer of each model. However, just as Dropout regularises the model by reducing its complexity, training on only a subset of the feature dimensions may also reduce each model's access to information in the input features. This may result in a sub-optimal performance for each system. Unlike the Monte Carlo Dropout ensemble generation method in Section 3.1.3, each member of the ensemble here is trained separately. Also, unlike Monte Carlo Dropout,

the feature subsampling method does not require all members of the ensemble to use the same topology.

3.2.3 Random initialisation

NN training is a highly non-convex problem when the NN has at least one hidden layer [25]. Many local optima may exist. Because of this, the final trained model behaviour is often sensitive to the initial parameter initialisation [68]. pre-training methods [9, 68] can be used to initialise the model parameters within a good basin of attraction, and avoid convergence to poor local optima. This sensitivity to initialisation can allow multiple sets of model parameters to be generated. This can simply be achieved by randomly sampling multiple sets of initial model parameters from a prior, and training each model separately until convergence. However, all members of the ensemble here have the same topology. One possible prior to sample model parameters from is (2.93) and (2.94). As is discussed in Section 2.4.3, this form of prior, proposed in [50], can initialise the NN parameters within a dynamic range that is conducive for gradient-based training. When starting from different random parameter initialisations, it is possible that the models may converge to different local optima after training, and may therefore exhibit diversity in their behaviours. However, this ensemble method is limited to only considering sets of model parameters that reside near local optima of the training criterion. When using MAP criteria, such as cross-entropy or sequence-level $\mathcal{F}_{\text{MMI}}$ with regularisation, this corresponds to models near the local maxima of $p\left(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D}\right)$.

3.2.4 Intermediate model iterations within a single training run

The methods discussed in Sections 3.2.1, 3.2.2, and 3.2.3, generate an ensemble by separately training multiple systems. This can lead to a high computational cost during training. A less expensive ensemble generation method is to use the models at intermediate training iterations of a single run of training as the members of the ensemble [131]. When using an iterative training algorithm, the training iterations explore a limited space of model parameters, eventually converging to near a local optimum. Furthermore, when using a mini-batch method, such as SGD, the noise of the stochastic update can lead to a wider model parameter space exploration. An ensemble constructed from these intermediate models may be able to capture the diversity of the explored model parameter space. This approach is similar to the Markov chain Monte Carlo ensemble generation method discussed in Section 3.1.4. However, unlike the Markov chain Monte Carlo method, no additional Gaussian noise is added to the gradient here. As such, it cannot be proven using the Fokker-Planck equation

that the collection of intermediate model iterations here converge to the true model parameter posterior, $p(\Phi|\mathcal{M}, \mathcal{D})$. Furthermore, by not adding addition Gaussian noise, the space of model parameters that is explored may be reduced, thereby leading to less diversity between the models. Similarly to the Markov chain Monte Carlo method, the consecutive models here can be expected to be highly correlated, and therefore have fairly similar behaviours to each other. This may further limit the ensemble diversity.

3.3 Sources of diversity in the structured models of ASR

The methods of obtaining diverse systems described in Section 3.2 are generic, and can be applied across a wide range of machine learning tasks. Section 2.2 describes how HMM-based systems in ASR are composed of separate acoustic, language, and alignment models, which capture attributes of speech data at different acoustic scales. When designing an ASR system, choices also need to be made about the feature representation, set of sub-word units, and set of state clusters. As such, it may be possible to introduce diversity between the system behaviours by allowing for differences between each of these models and design aspects. It may also be possible to obtain a richer ensemble by using multiple forms of diversities. Although the ensemble methods discussed here are centred around an HMM-based system, these methods are also widely applicable to other ASR system architectures. It is even possible to construct an ensemble encompassing different ASR system architectures, such as by combining NN-HMM, GMM-HMM, and CTC systems.

3.3.1 Feature diversity

Section 3.2.2 has described an ensemble generation method where multiple systems are trained, each using only a subset of the input feature dimensions. However, this approach may result in each individual system being suboptimal. In ASR, there exists a wide range of possible feature types, described in Section 2.2.6. The different feature types may represent and emphasise different aspects of the audio information. This diversity of input representations can be utilised to create an ensemble of systems, each built on a different feature type [28, 126]. Unlike the feature subsampling method, the systems here have access to all of the information captured within their respective feature types.

However, when using different hand-crafted feature representations, each feature extractor is manually designed. It may therefore be difficult to increase the ensemble size with new feature extractor designs. The NN feature extractor is a highly flexible model. Using this,

it may be possible to obtain a wide diversity of feature representations [28]. The methods described in Sections 3.1 and 3.2 can be used to obtain multiple diverse NN feature extractors.

3.3.2 Acoustic model diversity

Many different acoustic model topologies have been proposed [12, 76, 121, 147], each with different capacities and behaviours. Even within a single topology, it is possible to obtain a wide diversity of behaviours from different sets of model parameters. Ensembles with different sets of acoustic model parameters have been examined in [154], while ensembles that use different acoustic model topologies have been examined in [33]. Using only different sets of model parameters, all with the same topology, may limit the diversity that can be captured by the ensemble. This is in some ways analogous to how the constrained Bayesian inference of (3.2) does not allow all possible acoustic model topologies to be explored, leading to a biased estimate of the full Bayesian inference of (3.1). However, when considering different acoustic model topologies, each topology is often manually designed. This may make it difficult to increase the ensemble size with new model topologies. Using an ensemble with different sets of model parameters can allow for large ensembles, generated using the methods described in Sections 3.1 and 3.2.

3.3.3 State cluster diversity

As is discussed in Section 2.2.4, state clustering is often used to allow context-dependent modelling, while limiting the number of parameters. Instead of using a single set of state clusters defined by a single decision tree, an ensemble can be generated, such that each member uses a different set of state clusters, defined by a different decision tree [135]. In a hybrid NN-HMM system, the state clusters define the NN output classes, as is described in Section 2.2.5. When each system uses a different set of state clusters, the acoustic models aim to discriminate between different sets of output classes [172]. This may encourage the systems to develop highly diverse behaviours.

When using decision trees to perform state clustering, multiple decision trees can be obtained by inserting randomness into the tree building process. The random forest method is one possible way to generate multiple decision trees for the ensemble [34]. The standard method of training a decision tree is to iteratively choose greedy splits using (2.33) [169]. Instead, the random forest method first forms a list of n-best splits at each iteration, $\left\{\mathcal{T}_1^{(v)+1}, \cdots, \mathcal{T}_n^{(v)+1}\right\}$, as is shown in Figure 3.1. Here again, $\mathcal{T}^{(v)+1}$ represents the tree at the vth iteration with one additional split. Then, a split is sampled uniformly from the n-best

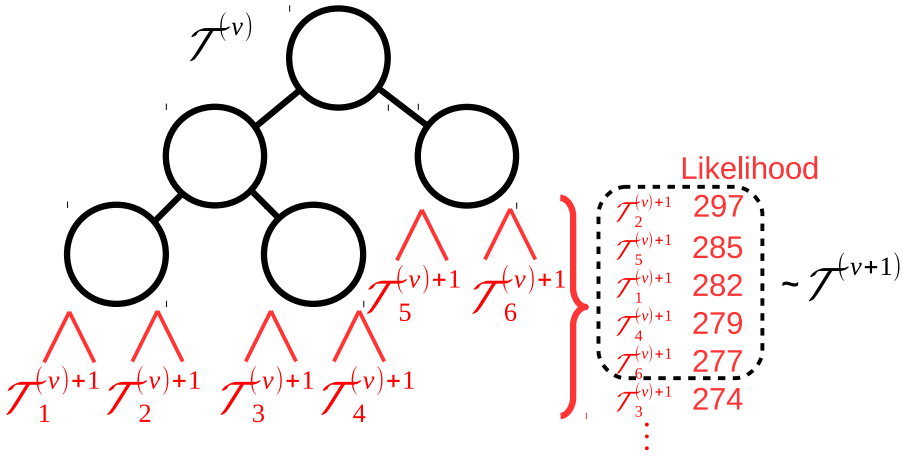


Fig. 3.1 Random forest decision tree sampling iteration. At each iteration, all possible next splits, $\mathcal{T}_i^{(v)+1}$, are listed in order of likelihood, and the next split, $\mathcal{T}^{(v+1)}$, is sampled uniformly from the n-best possible next splits. Although a tree from only a single root is shown here, the next splits are listed across all of the tree roots in practice.

list to give the tree at the next training iteration,

$$\mathcal{T}^{(v+1)} \sim \mathcal{U}\left\{\mathcal{T}_1^{(v)+1}, \cdots, \mathcal{T}_n^{(v)+1}\right\}. \tag{3.26}$$

Running the random forest decision tree building process multiple times with different random seeds results in multiple different decision trees.

Other ensemble generation methods have also been investigated that explicitly train the decision trees to be different. Work in [15] adapts Adaboost to the tree building process, by weighing the contribution of each frame to the likelihood computation, according to how well it is modelled by existing decision trees. Work in [162] jointly trains multiple decision trees, by interpolating an additional term into the training criterion that seeks to maximise the entropy of the observation likelihoods of the states in the intersection of the decision trees.

3.3.4 Sub-word unit diversity

Section 2.2.2 has discussed how words are often decomposed into sequences of sub-word units in ASR. The traditional approach is to use phones as the sub-word units. However, for each new language, building a phonetic system requires linguistic expertise to produce a dictionary of pronunciation decompositions for all of the words. This expertise is often expensive to obtain. An alternative is to decompose words orthographically, into sequences of graphemes [83]. A graphemic dictionary does not require any linguistic expertise to obtain, as the words are decomposed exactly as how they are spelt. This approach is suitable for

languages where there is a close correspondence between the written and spoken forms of words. Several works have investigated the possibility of automatically deriving a phonetic dictionary from a graphemic dictionary using a G2P model, to overcome the need for linguistic expertise [81, 129].

Table 3.1 Phonetic and graphemic decompositions of English words. Grapheme and phone sequences of the same word can have different lengths. There can be multiple phone sequences for each word, but only one grapheme sequence.

Word	Graphemes	Phones
the	/t/ /h/ /e/	/th/ /ax/
		/d/ /iy/
know	/k/ /n/ /o/ /w/	/n/ /ow/
B.B.C.	/b;DB/ /b;DB/ /c;DB/	/b/ /iy/ /b/ /iy/ /s/ /iy/

Rather than building an ASR system using a manually or automatically derived phonetic dictionary, it is also possible to build a system directly on a graphemic dictionary [45, 83]. However, for many languages, there may not be a close grapheme-to-phone relationship [83]. English is one such language where this relationship is highly variable. Table 3.1 gives examples of this variability in the sub-word unit decompositions for English. This example shows that the lengths of the phonetic and graphemic decompositions can vary. When relating the graphemic to the phonetic decompositions, there can be cases of omissions (/k/ /n/→/n/) and recombinations (/t/ /h/→/th/), where a sequence of multiple graphemes is related to a single phone. There can also be cases where a single grapheme relates to a sequence of multiple phones (/b;BD/→/b/ /iy/). Furthermore, unlike a graphemic dictionary, a phonetic dictionary can contain multiple decompositions for each word, capturing different possible pronunciations. As a result, graphemes often have a wider variety of possible acoustic realisations than phones [150]. The use of context-dependent sub-word units can partially alleviate this [150]. In the example in Table 3.1, a ";DB" in the grapheme indicates that the character is followed by a ".". Such information can also be incorporated into the graphemic dictionary to capture some acoustic variability. However, a greater demand is often still placed on the graphemic acoustic model to capture the wider variety of acoustic realisations.

This difference in word decompositions can lead to different behaviours between phonetic and graphemic systems. Furthermore, having different sets of sub-word units in turn requires the sets of state clusters to also be different. An ensemble of systems built using these different sets of sub-word units can exhibit a large diversity of behaviours, leading to performance gains through combination [150].

3.3.5 Language model diversity

It is also possible to have a diversity of language models. The methods in Sections 3.1 and 3.2 can be applied not only to the acoustic model, but also to generate an ensemble of language models. These techniques are directly applicable when using NN-based language models, such as an RNN language model. The different language models can be combined by interpolating their probabilities [74], for example as

$$P\left(\boldsymbol{\omega}_{1:L}\middle|\widehat{\boldsymbol{\Phi}}\right) = \lambda P\left(\boldsymbol{\omega}_{1:L}|\boldsymbol{\Phi}^{n\text{-gram}}\right) + (1-\lambda)\, P\left(\boldsymbol{\omega}_{1:L}\middle|\boldsymbol{\Phi}^{\text{RNN}}\right), \tag{3.27}$$

where $\widehat{\boldsymbol{\Phi}}$ represents the combined language model and the interpolation weight satisfies $0 \leq \lambda \leq 1$. This interpolation represents a combination at the language model level. It is also possible to perform combination at the hypothesis level when using different language models, as is described in Section 3.4.1.

As is described in Section 2.2.1, RNN language models are often only used to rescore a finite number of hypotheses, represented within an n-best list or a lattice. Rescoring can only change the rank order of existing hypotheses, but cannot introduce any new hypotheses. As such, the amount of diversity that can be introduced by rescoring a lattice with different RNN language models may be limited.

3.4 Ensemble combination

Having generated an ensemble, the systems then need to be combined together to perform recognition. This section discusses combining the systems at the hypothesis, frame, and feature levels. The form of combination influences both the computational cost of performing recognition and the forms of diversities that the ensemble is allowed to have.

3.4.1 Hypothesis level

The systems can be combined over hypothesis posteriors, as is shown in Figure 3.2. One possible method is to take a weighted average of the hypothesis posteriors,

$$P\left(\boldsymbol{\omega}_{1:L}\middle|\mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) = \sum_{m=1}^{M} \lambda_m P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right), \tag{3.28}$$

where $\widehat{\boldsymbol{\Phi}} = \{\boldsymbol{\Phi}^m \quad \forall m\}$ represents the ensemble, and the interpolation weights satisfy $0 \leq \lambda_m \leq 1$ and $\sum_m \lambda_m = 1$. From here on, the dependence on the model topology and system design, $\mathcal{M}$, is omitted for brevity. As is discussed in Section 3.1, if the systems

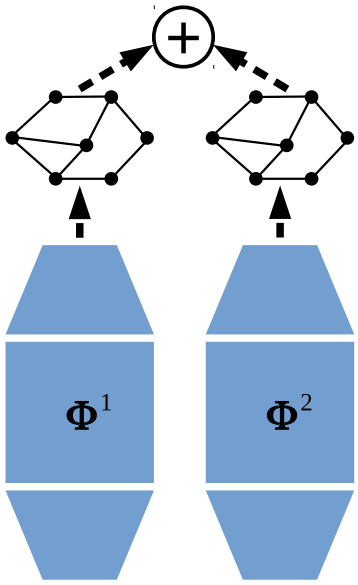


Fig. 3.2 Hypothesis-level combination of MBR combination decoding, CNC, or ROVER between two systems. For each system, data is fed through an acoustic model and a lattice is generated.

are sampled from the true posterior of $p\left(\boldsymbol{\Phi}|\mathcal{M},\mathcal{D}\right)P\left(\mathcal{M}|\mathcal{D}\right)$ or the interpolation weights are computed according to importance sampling using (3.4), then this combination method can be used to obtain an unbiased Monte Carlo estimate to Bayesian inference of the word sequence hypothesis. However, these requirements are often not satisfied in practice, due to the computational difficulties involved.

The combined hypothesis posteriors of (3.28) can be used together with MBR decoding in (2.9), to perform MBR combination decoding [163],

$$\boldsymbol{\omega}^{*}=\arg\min_{\boldsymbol{\omega}'}\sum_{\boldsymbol{\omega}}\mathcal{L}\left(\boldsymbol{\omega},\boldsymbol{\omega}'\right)\sum_{m=1}^{M}\lambda_{m}P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T},\boldsymbol{\Phi}^{m}\right). \tag{3.29}$$

As with MBR decoding, different forms of risk functions, $\mathcal{L}$, can be used. A word-level Levenstein distance of (2.7) can be used to obtain a combined hypothesis that minimises the expected WER. The risk function of (2.10) can be used to obtain the combined MAP hypothesis, with the minimum hypothesis classification error rate, or SER.

Section 2.1 has discussed how MBR decoding can be simplified, using a CN. This first simplifies the lattice of competing hypotheses into aligned confusion sets of competing words, with the approximation of (2.11). The MBR hypothesis can then simply be found by choosing the most probable word within each confusion set, as in (2.12). CN decoding can be applied to an ensemble. Rather than taking a combination over the hypothesis posteriors

in (3.28), a alternative method is to combine over the posteriors of each word,

$$P\left(\boldsymbol{\omega}_{1:L}\middle|\mathbf{O}_{1:T},\widehat{\boldsymbol{\Phi}}\right)=\prod_{l=1}^{L}\sum_{m=1}^{M}\lambda_m P\left(\omega_l|\mathbf{O}_{1:T},\boldsymbol{\Phi}^m\right). \tag{3.30}$$

This combination method can be easily incorporated with CN decoding, in a scheme known as CN Combination (CNC) [37, 100]. Here, the word posteriors, $P\left(\omega_l|\mathbf{O},\boldsymbol{\Phi}^m\right)$, represent the confusion sets from each system, with alignment performed across all of the systems. The combination over word posteriors can be implemented by merging together the confusion sets across the systems. As with CN decoding, the MBR hypothesis of the combined posteriors of (3.30) can simply be found by choosing the most probable word within each of the combined confusion sets,

$$\boldsymbol{\omega}^*=\arg\max_{\omega_1}\sum_{m=1}^{M}\lambda_m P\left(\omega_1|\mathbf{O}_{1:T},\boldsymbol{\Phi}^m\right),\cdots,\arg\max_{\omega_L}\sum_{m=1}^{M}\lambda_m P\left(\omega_L|\mathbf{O}_{1:T},\boldsymbol{\Phi}^m\right). \tag{3.31}$$

An even simpler hypothesis-level combination scheme is Recogniser Output Voting Error Reduction (ROVER) [38]. In ROVER, the 1-best hypotheses of the systems are first aligned to each other. The combined hypothesis is then selected using majority voting,

$$\omega_l^*=\arg\max_{\omega_l}\left\{\sum_{m=1}^{M}\left[u\frac{\delta\left(\omega_l,\omega_l^{m*}\right)}{M}+(1-u)\,\mathrm{conf}\left(w_l|\boldsymbol{\Phi}^m\right)\right]\right\}, \tag{3.32}$$

where ω_l^{m*} is the lth word in the 1-best hypothesis of the mth system after alignment, $\mathrm{conf}\left(w_l|\boldsymbol{\Phi}^m\right)$ is a confidence score for word w_l provided by the mth system, and u is an interpolation weight between the word counts and word confidence scores, satisfying $0\leq u\leq 1$. One possible choice for the form of the confidence score is to use the word posteriors, $\mathrm{conf}\left(w_l|\boldsymbol{\Phi}^m\right)=P\left(\omega_l|\mathbf{O}_{1:T},\boldsymbol{\Phi}^m\right)$ [37]. More information about the uncertainty that each system has about the hypothesis can be taken into account by considering an n-best list of hypotheses from each system, instead of only the 1-best. ROVER combination can be made to resemble CNC, by using $u=0$, $\mathrm{conf}\left(w_l|\boldsymbol{\Phi}^m\right)=P\left(\omega_l|\mathbf{O}_{1:T},\boldsymbol{\Phi}^m\right)$, and by considering multiple competing hypotheses from each system [37, 100].

These hypothesis-level combination methods are general, in the sense that they place few restrictions on the forms of diversities that the ensemble is allowed to have. ROVER combination does not place any restrictions on the systems. MBR combination decoding and CNC respectively only require valid hypothesis and word posteriors to be obtainable from the systems.

3.4.2 Frame level

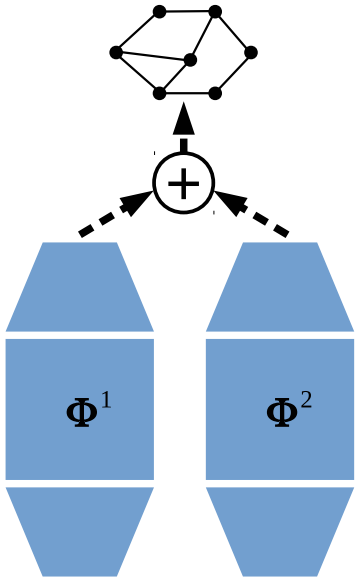


Fig. 3.3 Frame-level combination over per-frame posteriors or likelihoods between two systems. Data is fed through each acoustic model, but only a single lattice is required.

Hypothesis-level combination can be computationally expensive, as it requires a separate decoding run for each system within the ensemble. Each decoding run requires a separate lattice to be processed, using the recognition process described in Section 2.5.3. The ensemble can instead be combined at the frame level, illustrated in Figure 3.3. This only requires a single decoding lattice to be processed for the whole ensemble.

Frame-level combination can be performed over the state cluster posteriors,

$$P\left(s\middle|\mathbf{o}_t,\hat{\mathbf{\Phi}}\right)=\sum_{m=1}^{M}\lambda_m P\left(s|\mathbf{o}_t,\mathbf{\Phi}^m\right), \tag{3.33}$$

where the interpolation weights satisfy $0\leq\lambda_m\leq 1$ and $\sum_m\lambda_m=1$. Similarly to (2.46), the combined scaled observation likelihoods can be obtained from the combined state cluster posteriors as

$$\mathcal{A}\left(\mathbf{o}_t,s,\hat{\mathbf{\Phi}}\right)=\frac{P\left(s\middle|\mathbf{o}_t,\hat{\mathbf{\Phi}}\right)}{P\left(s\right)}. \tag{3.34}$$

The hypotheses can then be computed from these combined scaled observation likelihoods using (2.47). This frame-level combination method requires state cluster posteriors from each of the systems. It can therefore readily be used with NN-based acoustic models. Alternatively,

it is also possible to perform combination directly over the scaled observation likelihoods,

$$\mathcal{A}\left(\mathbf{o}_t, s, \widehat{\mathbf{\Phi}}\right) = \sum_{m=1}^{M} \lambda_m \frac{P\left(s|\mathbf{o}_t, \mathbf{\Phi}^m\right)}{P\left(s\right)}, \tag{3.35}$$

where the interpolation weights here satisfy $\lambda_m \geq 0$. This method can be used to combine GMM and NN acoustic models [148]. A normalised hypothesis posterior can still be obtained even if the combined scaled observation likelihoods are not normalised, by computing the hypothesis posterior using (2.47).

The sum combinations shown here are only one of the many possible schemes. Several possible frame-level combination schemes, such as taking a product or max over the posteriors or scaled likelihoods [139], are discussed in [87]. When using a product combination, combining the ensemble over state cluster posteriors and scaled observation likelihoods are equivalent,

$$\mathcal{A}\left(\mathbf{o}_t, s, \widehat{\mathbf{\Phi}}\right) = \prod_{m=1}^{M} P^{\lambda_m}\left(\mathbf{o}_t|s, \mathbf{\Phi}^m\right) \tag{3.36}$$

$$= \frac{1}{\prod_{m=1}^{M} P^{\lambda_m}\left(s\right)} \prod_{m=1}^{M} P^{\lambda_m}\left(s|\mathbf{o}_t, \mathbf{\Phi}^m\right), \tag{3.37}$$

if $\sum_m \lambda_m = 1$.

Unlike hypothesis-level combination, these frame-level combination methods assume that all systems in the ensemble use the same set of state clusters. Section 3.3.3 has discussed the possibility of having a diversity of state cluster sets in an ensemble. Work in [157] demonstrates that such an ensemble can be combined at the hypothesis level. Frame-level combination can be generalised to allow for different sets of state clusters by performing combination over the logical context-dependent state scaled observation likelihoods [165]. When performing state clustering, the scaled observation likelihoods of all logical context-dependent states within the same state cluster are tied according to (2.29), re-expressed here for each system as

$$\mathcal{A}\left(\mathbf{o}_t, c, \mathbf{\Phi}^m\right) = \mathcal{A}\left(\mathbf{o}_t, s_c^m, \mathbf{\Phi}^m\right), \tag{3.38}$$

where s_c^m is the state cluster that the logical context-dependent state c belongs to when using the mth decision tree, described as (2.28). Frame-level combination with different sets of

state clusters can be performed as

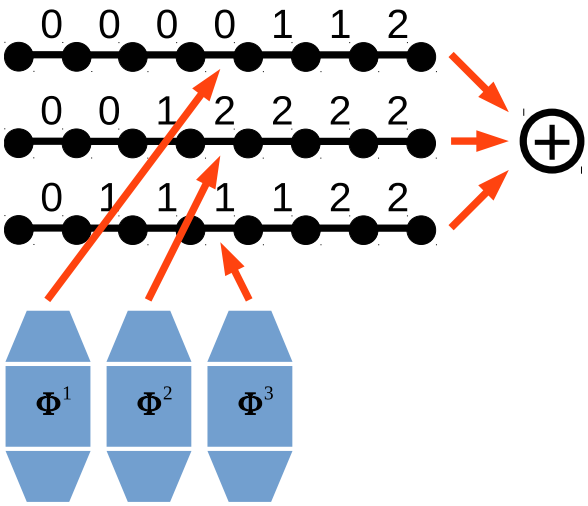
$$\mathcal{A}\left(\mathbf{o}_t, c, \widehat{\mathbf{\Phi}}\right) = \sum_{m=1}^{M} \lambda_m \mathcal{A}\left(\mathbf{o}_t, c, \mathbf{\Phi}^m\right) \tag{3.39}$$

$$= \sum_{m=1}^{M} \lambda_m \frac{P\left(s_c^m | \mathbf{o}_t, \mathbf{\Phi}^m\right)}{P\left(s_c^m\right)}. \tag{3.40}$$

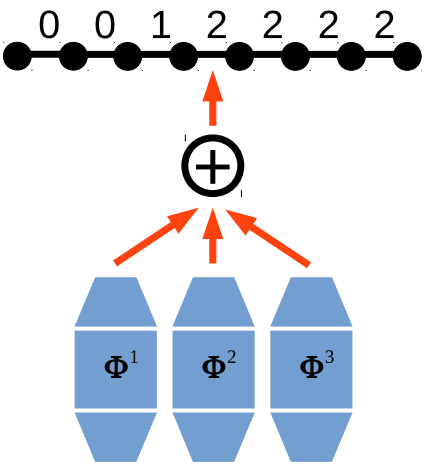
Using this combination method, it can be seen that all logical context-dependent states, c, that are mapped to the same group of M state clusters, $\left\{s^1, \cdots, s^M\right\}$, will have the same scaled observation likelihood. Therefore, it is possible to define a set of intersect state clusters, $\widehat{s}$, such that all logical context-dependent states within each intersect state are mapped to the same M state clusters [165]. This set of intersect states is the set of state clusters formed by the Cartesian product of all of the decision trees, ignoring all state clusters of the product that do not contain any logical context-dependent states. The set of intersect states represents the effective phonetic resolution of the ensemble, as this is the set of unique scaled observation likelihoods that can be computed using the ensemble. The scaled observation likelihoods therefore only need to be computed over these intersect states, instead of the logical context-dependent states, without any loss of generality.

Hypothesis-level combination requires data to be fed through each of the separate acoustic models, with a computational cost that scales as $\mathcal{O}\left(M\right)$, and separate decoding lattices to be processed, again with a computational cost that scales as $\mathcal{O}\left(M\right)$. When performing frame-level combination, the combined scaled observation likelihoods are used to generate only a single decoding lattice for the whole ensemble. As such, frame-level combination is less computationally expensive to use when performing recognition than hypothesis-level combination. However, data still needs to be fed through each of the acoustic models separately. Therefore, frame-level combination still has a computational cost scaling as $\mathcal{O}\left(M\right)$. Also, since only a single decoding lattice is processed for the whole ensemble, all systems are required to use the same language model, set of sub-word units, and HMM topology.

Frame-level combination requires that all system traverse the same state sequence, illustrated in Figure 3.4b. As opposed to this, hypothesis-level combination can allow each system to have different state sequences, illustrated in Figure 3.4a. Frame-level combination therefore requires time-synchronous state transitions and the same pronunciation variants between systems, which may limit the diversity that the ensemble can express. When constructing an ensemble to be used with frame-level combination, it may be important to train the systems to abide by these requirements. When training using the cross-entropy criterion of (2.80), this can be encouraged by using the same set of forced alignments for all systems.



(a) Asynchronous transitions with hypothesis-level combination. Each system is allowed to have different state transition times.



(b) Synchronous transitions with frame-level combination. All systems must have the same state transition times.

Fig. 3.4 Frame-level combination assumes that the state transitions of all systems are synchronous. Figure shows the traversal of a sequence of state clusters 0, 1, and 2.

This can also be done when the systems use different sets of state clusters, by mapping a common set of forced alignments to each of the different sets of state clusters [136].

3.4.3 Feature level

Although frame-level combination is less computationally expensive than hypothesis-level combination, it still requires data to be fed through each of the individual acoustic models. This computational cost can be further reduced using feature-level combination.

As is shown in Figure 3.5, an NN acoustic model with parameters $\boldsymbol{\Phi} = \{\boldsymbol{\Psi}, \boldsymbol{\Xi}\}$ can be interpreted as being composed of a first stage feature extractor with parameters $\boldsymbol{\Psi}$ and a second stage classifier with parameters $\boldsymbol{\Xi}$. The first stage feature extractor maps from the input features, $\mathbf{o}_t$, to features in the hidden representation of $\acute{\mathbf{o}}_t$, similarly to (2.65),

$$\acute{\mathbf{o}}_t = \boldsymbol{\varphi}\left(\mathbf{o}_t | \boldsymbol{\Psi}\right). \tag{3.41}$$

Here, for simplicity, the feature extractor is illustrated to operate on a per-frame basis. A more general sequence-to-sequence feature extractor, as in (2.65), can also be used. The second stage classifier produces state cluster posteriors, $P\left(s | \acute{\mathbf{o}}_t, \boldsymbol{\Xi}\right)$, given the features $\acute{\mathbf{o}}_t$.

Rather than performing frame-level combination over the state cluster posteriors or scaled observation likelihoods, an ensemble can instead be combined over the features, $\acute{\mathbf{o}}_t^m$,

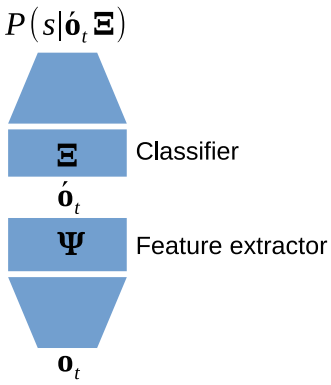


Fig. 3.5 A neural network acoustic model is composed of a feature extractor and a classifier. The feature extractor, with parameters Ψ, takes input features $\mathbf{o}_t$ and projects them to the features $\acute{\mathbf{o}}_t$. The classifier, with parameters Ξ, takes the projected features and produces a state cluster posterior.

computed from each system,

$$\acute{\mathbf{o}}_t^m = \varphi\left(\mathbf{o}_t | \Psi^m\right). \tag{3.42}$$

Using this feature-level combination, data only needs to be fed through the classifier, Ξ, once for the whole ensemble, as is illustrated in Figure 3.6. It can therefore be less computationally expensive to perform combination at the feature level than the frame level.

However, unlike the hypothesis or frame-level combination methods, the features that are combined in feature-level combination need not have a probabilistic interpretation. Therefore, using a weighted average, as in (3.28) and (3.33), or one of the other combination methods described in [87], may not make sense for feature-level combination. A more general combination method is to concatenate the features from the different systems together,

$$\widehat{\mathbf{o}}_t = \left[\acute{\mathbf{o}}_t^1 \quad \cdots \quad \acute{\mathbf{o}}_t^M\right], \tag{3.43}$$

where $\widehat{\mathbf{o}}_t$ are the combined features. These combined features are then fed through the common classifier.

Frame-level combination over the state cluster posteriors using (3.33) can in fact be viewed as a specific instance of feature-level combination. Here, the extracted features are the state cluster posteriors,

$$\acute{o}_{st}^m = \varphi_s\left(\mathbf{o}_t | \Psi^m\right) \tag{3.44}$$

$$= P\left(s | \mathbf{o}_t, \Phi^m\right), \tag{3.45}$$

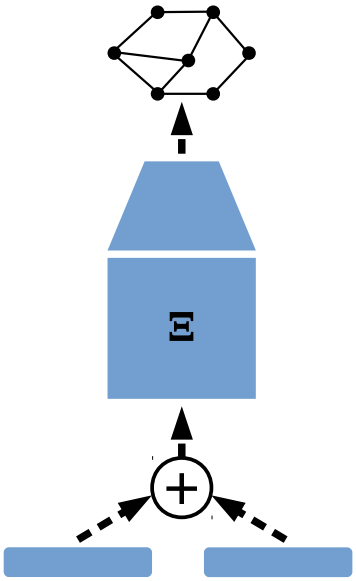


Fig. 3.6 Feature-level combination. Combination can be performed as a concatenation of the feature vectors. Data only needs to be fed through a single acoustic model classifier, and only a single lattice is required.

with $\boldsymbol{\Psi}^m = \boldsymbol{\Phi}^m$, and the classifier is

$$P\left(s|\hat{\mathbf{o}}_t, \boldsymbol{\Xi}\right) = \sum_{m=1}^{M} \lambda_m \acute{o}_{st}^m, \tag{3.46}$$

such that the classifier parameters are the interpolation weights, $\boldsymbol{\Xi} = \{\lambda_m \quad \forall m\}$. More generally, the classifier can take the form of a linear matrix multiplication [33], or even be an NN.

The features that are combined in (3.43) do not need to be state cluster posteriors at the outputs of the NN of (3.45), but can be the activations at any of the hidden layers. For each member of the ensemble, data only needs to be fed through each of the separate feature extractors up to where the features to be combined, $\acute{\mathbf{o}}_t^m$, are produced. Combining the features at a lower hidden layer therefore reduces the computational cost of combination. It is even possible to perform feature-level combination over different input features, $\mathbf{o}_t^m$. Section 3.3.1 has discussed the possibility of obtaining ensemble diversity by using a different set of features for each member of the ensemble. Feature-level combination can be used to combine such an ensemble in an efficient manner [28, 126].

3.5 Feature diversity using an echo state network

Section 3.3.1 has discussed how a diversity of feature representations can be included in an ensemble by using different hand-crafted features. However, the size of such an ensemble is limited by the possible types of hand-crafted features that are available. Instead, an ensemble of NN feature extractors can also be used to obtain feature diversity. Each feature extractor projects the features according to (2.65), repeated here for each member of the ensemble as

$$\acute{\mathbf{o}}_t^m = \boldsymbol{\varphi}_t \left(\mathbf{O}_{1:T} | \boldsymbol{\Phi}^m \right), \tag{3.47}$$

where $\boldsymbol{\Phi}^m$ are the parameters of the mth NN feature extractor. The methods discussed in Sections 3.1 and 3.2 can be used to generate an ensemble of NN feature extractors.

Section 3.1 has discussed how an ensemble can be generated by sampling sets of model parameters from the true model parameter posterior, $p\left(\boldsymbol{\Phi}|\mathcal{D}\right)$, or an approximate posterior, $q\left(\boldsymbol{\Phi}|\mathcal{D}\right)$. Here, the dependence on the model topology and system design, $\mathcal{M}$, is omitted for brevity. These methods can be used to generate an ensemble of NN feature extractors, by sampling multiple sets of model parameters. The Markov chain Monte Carlo method discussed in Section 3.1.4 can be used to sample sets of model parameters from the true model parameter posterior,

$$\boldsymbol{\Phi}^m \sim p\left(\boldsymbol{\Phi}|\mathcal{D}\right). \tag{3.48}$$

However, this method can be computationally expensive, as many samples may be required to obtain a diverse ensemble, because the consecutive samples may be highly correlated. Furthermore, each sample generated using this method requires a mini-batch of training data to be processed. This computational cost can be reduced by instead sampling sets of model parameters from a prior [140],

$$\boldsymbol{\Phi}^m \sim p\left(\boldsymbol{\Phi}\right). \tag{3.49}$$

Each set of NN feature extractor model parameters are sampled without knowledge of the training data. This method is referred to as random projections. However, although it may be less computationally expensive to sample the parameters from a prior, these sampled NN feature extractors may not be optimal.

Each sampled NN feature extractor projects the features according to (3.47). These can then be combined in a computationally efficient manner using feature-level combination of (3.43), repeated here as

$$\widehat{\mathbf{o}}_t = \begin{bmatrix} \acute{\mathbf{o}}_t^1 & \cdots & \acute{\mathbf{o}}_t^M \end{bmatrix}. \tag{3.50}$$

The combined features, $\widehat{\mathbf{o}}_t$, can also be interpreted as having been projected by a single combined feature extractor,

$$\widehat{\mathbf{o}}_t = \boldsymbol{\varphi}_t \left(\mathbf{O}_{1:T} \middle| \widehat{\boldsymbol{\Phi}} \right) \tag{3.51}$$

composed of all of the individual feature extractors,

$$\widehat{\boldsymbol{\Phi}} = \left\{ \boldsymbol{\Phi}^1, \cdots, \boldsymbol{\Phi}^M \right\}. \tag{3.52}$$

Since each set of NN feature extractor model parameters are sampled from a prior, according to (3.49), the combined feature extractor model parameters can also be interpreted as having been sampled from a prior,

$$\widehat{\boldsymbol{\Phi}} \sim p\left(\widehat{\boldsymbol{\Phi}} \right), \tag{3.53}$$

which can be factorised as

$$p\left(\widehat{\boldsymbol{\Phi}} \right) = \prod_{m=1}^{M} p\left(\boldsymbol{\Phi}^m \right), \tag{3.54}$$

where $p\left(\boldsymbol{\Phi}^m \right)$ are priors from which each individual set of NN feature extractor parameters are sampled from, according to (3.49).

This thesis examines using the Echo State Network (ESN) [73] as a possible method of sampling random feature extractors from a prior. Before current RNN topologies and training techniques [70, 152, 153] were proposed, it was generally considered difficult to effectively train RNNs, primarily because of the issues relating to vanishing and exploding gradients [111]. The ESN was proposed as a possible method of overcoming this training difficulty. The ESN is a single RNN layer, whose parameters are randomly initialised and left untrained. A single linear layer would then be trained on the random projection outputs of the ESN. The motivation for this use of the ESN comes from Cover's separability theorem [26], which states that the probability that a linear separating hyper-plane exists in a binary classification problem increases with the dimension of a nonlinear random projection of the feature space. As such, if the projection dimension is large, then it is highly probable that the ESN random projections are linearly separable.

This thesis proposes to use the ESN as a method for sampling random feature projections. The outputs of these random projections are fed into a common classifier, using the feature-level combination method discussed in Section 3.4.3. Therefore, unlike the standard approach of using a linear output layer with the ESN [73], this thesis proposes to train a DNN classifier on the ESN projections.

The ESN is composed of a single RNN layer,

$$\mathbf{z}_t = \mathbf{W}^{\mathrm{I}}\mathbf{o}_t + \mathbf{W}^{\mathrm{R}}\widehat{\mathbf{o}}_{t-1} + \mathbf{b} \tag{3.55}$$

$$\widehat{\mathbf{o}}_t = \tanh\left(\mathbf{z}_t\right), \tag{3.56}$$

where $\mathbf{W}^{\mathrm{I}}$, $\mathbf{W}^{\mathrm{R}}$, and $\mathbf{b}$ are the input matrix, recurrent matrix, and bias respectively. The ESN projections, $\widehat{\mathbf{o}}_t$, can be interpreted as the concatenation of multiple separate features, $\widehat{\mathbf{o}}_t = \begin{bmatrix} o_t^1 & \cdots & o_t^M \end{bmatrix}$, where each separate feature can be viewed as a new member of the ensemble. A single ESN can be used to generate a large number of feature projection samples, equal to the ESN projection dimension. As such the number of projection samples, M, can be interpreted as the ensemble size. The recurrent nature of the ESN may allow it to map information about the temporal context into the projections. Therefore, increasing the projection dimension beyond the dimension of $\mathbf{o}_t$ may still provide new information. A potential limitation of using the ESN to sample feature representations is that many of the ESN parameters are shared across the feature extractor samples. This may limit the diversity of the projected feature behaviours.

When using an ESN, the combined feature extractor parameters are

$$\widehat{\mathbf{\Phi}} = \left[\mathbf{W}^{\mathrm{I}}, \mathbf{W}^{\mathrm{R}}, \mathbf{b}\right]. \tag{3.57}$$

These parameters are randomly initialised and left untrained. This is equivalent to sampling a combined feature extractor, as in (3.53). Since the ESN parameters are not trained, it is important to place constraints on the parameters to allow for stability in the recurrent projections [73]. The ESN has a recurrent topology, and therefore may be able to retain information from past time steps. However, this recurrent memory may also lead to instability. Several works have attempted to assess the stability of an ESN, using either the largest absolute eigenvalue of $\mathbf{W}^{\mathrm{R}}$, known as the spectral radius,

$$\varpi\left(\mathbf{W}^{\mathrm{R}}\right) = \max\left|\mathrm{eig}\left(\mathbf{W}^{\mathrm{R}}\right)\right|, \tag{3.58}$$

or the largest singular value of $\mathbf{W}^{\mathrm{R}}$ [73, 109, 167]. The definition of a stable ESN is one where the influence of past inputs and the initial projection, $\widehat{\mathbf{o}}_0$, diminish with time [73]. However, these stability bounds are often loose or assume a linear ESN activation function, and satisfying them can lead to a rapid decay in the ESN memory [167]. In this thesis, ϖ is used as an approximate measure of the rate of memory decay, and stability is tested empirically.

Another measure of the ESN behaviour used in this thesis is the nonlinearity scale, defined as

$$\zeta\left(\mathbf{W}^{\mathrm{I}}, \mathbf{b}\right) = \sqrt{\dim\left(\mathbf{o}_t\right) \times \mathrm{var}\left\{\mathbf{W}^{\mathrm{I}}\right\} + \mathrm{var}\left\{\mathbf{b}\right\}}. \tag{3.59}$$

Here, the empirical variance of the elements in a matrix or vector is measured as

$$\mathrm{var}\left\{\mathbf{W}^{\mathrm{I}}\right\} = \frac{1}{MJ-1}\sum_{i=1}^{M}\sum_{j=1}^{J}\left(w_{ij}^{\mathrm{I}} - \frac{1}{MJ}\sum_{i'=1}^{M}\sum_{j'=1}^{J} w_{i'j'}^{\mathrm{I}}\right)^2, \tag{3.60}$$

where M and J are the numbers of rows and columns in $\mathbf{W}^{\mathrm{I}}$ respectively. This nonlinearity scale measures the approximate standard deviation of the pre-nonlinearity activations of $\mathbf{z}_t$ in (3.55), under the assumptions that there is no recurrence and the input, $\mathbf{o}_t$, has zero mean and unit variance. This approximate measure can indicate the degree to which the $\tanh$ activation function in (3.56) deviates from linearity.

In this thesis, the ESN parameters are sampled using a two-stage process to allow for control of the stability. First, an initial set of parameters, $\widehat{\mathbf{\Phi}}' = [\mathbf{W}^{\mathrm{I}\prime}, \mathbf{W}^{\mathrm{R}\prime}, \mathbf{b}']$, is sampled from a zero-mean and identity-covariance Gaussian probability density function,

$$\widehat{\mathbf{\Phi}}' \sim \mathcal{N}\left(\widehat{\mathbf{\Phi}}'; \mathbf{0}, \mathbf{I}\right). \tag{3.61}$$

Then, the final ESN parameters, $\widehat{\mathbf{\Phi}} = [\mathbf{W}^{\mathrm{I}}, \mathbf{W}^{\mathrm{R}}, \mathbf{b}]$, are obtained by rescaling these initial parameters to encourage recurrent stability,

$$\mathbf{W}^{\mathrm{R}} = \frac{\check{\varpi}}{\varpi\left(\mathbf{W}^{\mathrm{R}\prime}\right)}\mathbf{W}^{\mathrm{R}\prime} \tag{3.62}$$

$$\begin{bmatrix}\mathbf{W}^{\mathrm{I}} & \mathbf{b}\end{bmatrix} = \frac{\check{\zeta}}{\zeta\left(\mathbf{W}^{\mathrm{I}\prime}, \mathbf{b}'\right)}\begin{bmatrix}\mathbf{W}^{\mathrm{I}\prime} & \mathbf{b}'\end{bmatrix}. \tag{3.63}$$

The parameters are rescaled, to ensure that the measured $\varpi\left(\mathbf{W}^{\mathrm{R}}\right)$ and $\zeta\left(\mathbf{W}^{\mathrm{I}}, \mathbf{b}\right)$ values match the intended $\check{\varpi}$ and $\check{\zeta}$ values.

This method may allow for a wide variety of diverse feature projections to be sampled without a high computational cost. This ensemble of feature projections can then be combined in a computationally efficient manner using the feature-level combination method, discussed in Section 3.4.3.

3.6 Measuring diversity

When designing and generating an ensemble, it is useful to obtain an estimate of the diversity between the members of the ensemble, without explicitly evaluating the combined performance. These measures can be used to guide the ensemble design and diagnose issues that may arise. This section discusses several possible diversity measures.

3.6.1 Hypothesis diversity

The combined performance of an ensemble depends on the performance of the individual systems and the diversity between the behaviours of the systems [61]. The relationship between the diversity and combined performance can be seen when using the ROVER combination method, described in Section 3.4.1. This performs combination using the 1-best hypotheses from each system. Here, only the word counts are used, without any confidence scores. Table 3.2 illustrates an example of a possible outcome from performing ROVER combination across 3 systems.

Table 3.2 Systems can correct for each others' errors if they make different errors.

System 1:	the cat sat on a cat	2 errors
System 2:	a cat cat on a mat	3 errors
System 3:	the mat sat on a mat	2 errors
Combined:	the cat sat on a mat	1 error
Reference:	the cat sat on the mat	

The example shows that the systems are able to correct for each others' errors if the errors made by each system are different. The errors made by the systems can only differ if the hypotheses of the systems differ from each other. As such, one possible estimate of the diversity between the system behaviours is to measure the amount of disagreement between their hypotheses. This thesis proposes to measure the hypothesis disagreement between systems using the cross-WER,

$$\text{cross-WER} = \frac{1}{M\left(M-1\right)} \sum_{m=1}^{M} \sum_{n \neq m} \frac{1}{\sum\limits_{r'} L_{r'}^{n*}} \sum_{r} \mathcal{L}\left(\boldsymbol{\omega}_{r,1:L_r^{m*}}^{m*}, \boldsymbol{\omega}_{r,1:L_r^{n*}}^{n*}\right), \tag{3.64}$$

which is published by the author of this thesis in [155]. Here, $\boldsymbol{\omega}_{r,1:L_r^{m*}}^{m*}$ is the 1-best hypothesis of the rth utterance, obtained from the mth system. The risk function, $\mathcal{L}$, is the word-level Levenstein distance, defined in (2.7). This measures the word-level minimum edit distance between the 1-best hypotheses produced by the members of the ensemble, averaged over

all pairwise combinations of systems. This is related to the WER in (2.8), where here, the hypotheses of two systems are compared by treating one as the reference. An ensemble with a larger cross-WER has a greater diversity of 1-best hypotheses. This may indicate a greater diversity in the errors that are made by each system, which may allow more gains to be obtained in combination.

The concept of comparing the differences between the hypotheses of multiple systems is not new. Work in [49] computes a measure of the statistical significance of the difference between the performances of two systems, by comparing the differences in the errors made between the hypotheses of two systems and a reference.

The cross-WER measures diversity at the word level, and may be indicative of the ensemble performance when used with hypothesis-level combination methods. When using frame-level combination methods, it may be useful to measure the diversity, based on frame-level behaviour. An analogous diversity estimate to the cross-WER, but at the frame level, is to measure the difference between the 1-best per-frame state cluster classifications, referred to as the cross-Frame Error Rate (FER),

$$\text{cross-FER} = \frac{1}{TM\left(M-1\right)} \sum_{m=1}^{M} \sum_{n \neq m} \sum_{t=1}^{T} \left[1 - \delta\left(s_t^{m*}, s_t^{n*}\right)\right]. \tag{3.65}$$

When using an NN acoustic model, the 1-best state cluster classifications of the mth system can be computed as

$$s_t^{m*} = \arg\max_s P\left(s|\mathbf{o}_t, \boldsymbol{\Phi}^m\right). \tag{3.66}$$

Here, only substitution errors are considered. The cross-FER also only considers contributions to the diversity from the acoustic model and feature extractors. These are the same forms of diversities that are allowed when using frame-level combination.

3.6.2 Posterior diversity

The cross-WER and cross-FER only consider 1-best hypotheses or state cluster classifications. This does not take into account information about other competing hypotheses. A combination method such as MBR combination decoding in (3.29) utilises not only information about the 1-best hypotheses, but also information about the other competing hypotheses, captured within the hypothesis posteriors, $P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right)$. As such, it may be useful to compute a measure of the difference between these hypothesis posteriors of the systems. One such possible measure is the KL-divergence between the hypothesis posteriors, averaged over all

pairwise combinations of systems,

$$\text{hypothesis-KL} = \frac{1}{M\left(M-1\right)} \sum_{m=1}^{M} \sum_{n \neq m} \sum_{\boldsymbol{\omega} \in \mathbb{A}} P\left(\boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Phi}^{n}\right) \log \frac{P\left(\boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Phi}^{n}\right)}{P\left(\boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Phi}^{m}\right)}. \quad (3.67)$$

An implied sum over utterances is omitted here for brevity. This measure requires that the hypothesis posteriors produced by all systems have the same support over the span of competing hypotheses, $\boldsymbol{\omega}$. The hypothesis posteriors are often represented within a lattice, $\mathbb{A}$. The same support for all systems can be ensured by using the same lattice paths for all systems, rescored using the probabilities from the language, alignment, and acoustic models from each system. However, using the same lattice paths for all systems may underestimate the diversity between the systems. When using lattices, this KL-divergence can be computed through a forward-backward operation, using the expectation semi-ring, described in [94].

Combination can also be performed over the frame-level state cluster posteriors, using (3.33). At the frame-level, the diversity between the state cluster posteriors can similarly be measured using a KL-divergence,

$$\text{frame-KL} = \frac{1}{TM\left(M-1\right)} \sum_{m=1}^{M} \sum_{n \neq m} \sum_{t=1}^{T} \sum_{s \in \mathcal{T}} P\left(s | \mathbf{o}_{t}, \boldsymbol{\Phi}^{n}\right) \log \frac{P\left(s | \mathbf{o}_{t}, \boldsymbol{\Phi}^{n}\right)}{P\left(s | \mathbf{o}_{t}, \boldsymbol{\Phi}^{m}\right)}. \quad (3.68)$$

3.6.3 Phonetic decision tree diversity

As is discussed in Section 3.3.3, an ensemble can have a diversity of state cluster sets, by using multiple decision trees. When constructing such an ensemble, it may be useful to obtain an empirical measure of how different the decision trees are. One possible measure is the Tree Cluster Divergence (TCD) [14]. Here, the observation features within each state cluster are assumed to be Gaussian distributed. The diversity can be measured as the KL-divergence between the observation likelihoods of all state clusters of all pairs of decision trees,

$$\text{TCD} = \frac{1}{M\left(M-1\right)} \sum_{m=1}^{M} \sum_{n \neq m} \sum_{s^{m} \in \mathcal{T}^{m}} \sum_{s^{n} \in \mathcal{T}^{n}} \int \mathcal{N}\left(\mathbf{o}; \boldsymbol{\mu}_{s^{n}}, \boldsymbol{\Sigma}_{s^{n}}\right) \log \frac{\mathcal{N}\left(\mathbf{o}; \boldsymbol{\mu}_{s^{n}}, \boldsymbol{\Sigma}_{s^{n}}\right)}{\mathcal{N}\left(\mathbf{o}; \boldsymbol{\mu}_{s^{m}}, \boldsymbol{\Sigma}_{s^{m}}\right)} d\mathbf{o}, \quad (3.69)$$

where $\mathcal{T}^{m}$ is the set of state clusters defined by the mth decision tree. The Gaussian means and covariances can be estimated empirically from a forced alignment of the training data. This diversity measure does not require acoustic models to be trained, and is therefore computationally cheap to use. However, the tree cluster divergence assumes that the observation likelihoods are Gaussian distributed. In an actual system, the observation likelihoods

are instead often computed using a GMM in (2.34), or as scaled observations likelihoods computed from the state cluster posteriors of an NN in (2.46).

3.7 Summary

This chapter has reviewed several methods of generating an ensemble of diverse models. Multiple models can be sampled from a chosen posterior, and such a method can be related to Bayesian inference. However, these methods can be computationally expensive. Instead, systems can be generated with the aim of maximising diversity. Many different forms of diversities can be used in an ASR ensemble. When performing recognition, the ensemble can be combined at the hypothesis, frame, or feature levels. Performing combination further down the acoustic hierarchy is less computationally expensive, but places more restrictions on the allowed forms of ensemble diversities. This thesis has proposed using the ESN as a computationally cheap method of generating a diversity of feature representations. Several methods of measuring the diversity within an ensemble have also been proposed.

Chapter 4

Ensemble compression

The ensemble methods described in the previous chapter can often yield performance gains over using just a single system. However, the computational expense of performing recognition using multiple systems in the ensemble can present a hindrance, especially when deploying the ASR systems on devices with limited hardware resources. Similarly, a high computational cost can also arise when using a large single system.

This chapter looks at methods to reduce the computational cost of using a large system or an ensemble of systems. Section 4.1 discusses the low-rank matrix factorisation method, which can be used to reduce the number of parameters in each system. Section 4.2 describes an approach for compressing an ensemble by merging together the hidden layers, leaving only separate output layers as the members of the ensemble. This ensemble topology only requires data to be fed through the hidden layers once for the whole ensemble, but still requires data to be fed through each of the separate output layers. Section 4.3 then discusses several methods to further reduce the computational cost of performing recognition, by compressing the ensemble into a single system.

4.1 Low-rank matrix factorisation

The computational cost required to use an ensemble for recognition can be reduced by reducing the number of model parameters within each system. One possible method of achieving this within an NN model is to perform a low-rank factorisation of the NN matrices, which for a feed-forward DNN are used in the computation of (2.36) [124]. An NN matrix, $\mathbf{W}$, can be approximated by a factorised product of two matrices,

$$\widetilde{\mathbf{W}} = \mathbf{HL}, \tag{4.1}$$

where if $\mathbf{W}$ has dimension $I \times J$, then $\mathbf{L}$ has dimension $I \times K$ and $\mathbf{H}$ has dimension $K \times J$. The total number of parameters in $\widetilde{\mathbf{W}}$ is $IK + KJ$. There will therefore be fewer parameters in the $\widetilde{\mathbf{W}}$ than $\mathbf{W}$ if

$$K < \frac{IJ}{I+J}. \tag{4.2}$$

This is illustrated in Figure 4.1. Using factorised matrices in the NN can significantly reduce both the memory and number of computational operations needed when using the model. However, using a factorised weight matrix restricts the subspace dimension in which the hidden layer pre-activation projections can occupy. This may therefore be viewed as a regularisation technique, as it restricts the model capacity and the number of parameters. Low-rank factorisation may thus still allow the model to generalise well when the quantity of training data is small.

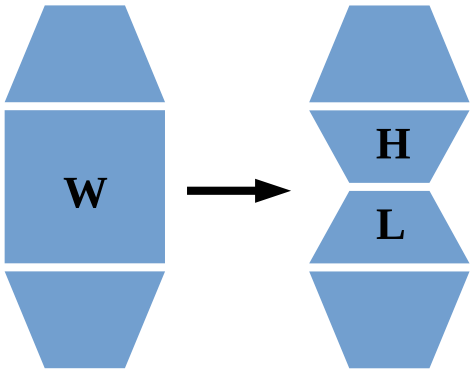


Fig. 4.1 Low-rank matrix factorisation of a single layer neural network. In general low-rank factorisation can be applied to multiple layers. The matrix of the linear transformation, $\mathbf{W}$, is factorised into two matrices, $\mathbf{L}$ and $\mathbf{H}$.

Low-rank factorisation of NN matrices was originally proposed in [124]. In [124], the NN model is both initialised and trained with low-rank matrices already incorporated in. Since there are fewer parameters, this model has a reduced modelling capacity. As such, it may not learn to perform as well as a full-rank model if there is a sufficient amount of training data available. Work in [164] improves upon this by proposing to start with a well trained full-rank model, and then replace the full-rank matrices with low-rank matrix approximations. This is achieved by performing singular value decomposition on the full-rank matrices, as

$$\mathbf{W} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}, \tag{4.3}$$

truncating off the smaller singular values to get $\mathbf{\Sigma}'$, $\mathbf{U}'$, and $\mathbf{V}'$, and then obtaining the low-rank matrix of (4.1) by setting

$$\mathbf{H} = \mathbf{U}' \quad \text{and} \quad \mathbf{L} = \mathbf{\Sigma}'\mathbf{V}'. \tag{4.4}$$

In this way, it is hoped that the low-rank approximation will perform similarly to the original full-rank model. Further fine-tuning can be performed on the low-rank model.

4.2 Multi-task ensemble

Another possible method of reducing the computational cost of performing recognition with an ensemble is to use an ensemble with a multi-task topology [136]. Multi-task learning in ASR has previously been proposed to simultaneously train a model on different but related tasks. Some examples of such related tasks include the classification of graphemes [22], monophones and surrounding contexts [130], and sets of state clusters from different decision trees [7]. Multi-task learning has also been proposed to train multi-lingual acoustic models [104]. Training on these multiple tasks has been found to improve generalisation on the main task [19]. One hypothesis of why training with the related tasks leads to improvements is that the model may develop a more general hidden representation, that is more widely applicable over a range of tasks [7].

The multi-task framework can also be used for ensemble compression. Inspiration for this method comes from the feature-level combination method in Section 3.4.3. As is discussed in Section 3.4.3, an NN acoustic model, $\mathbf{\Phi} = \{\mathbf{\Psi}, \mathbf{\Xi}\}$, can be decomposed into an initial feature extractor, $\mathbf{\Psi}$, followed by a classifier, $\mathbf{\Xi}$. An ensemble with a diversity of feature representations can be combined in a computationally efficient manner by using feature-level combination. Since the ensemble relies on a diversity of feature representations, only the feature extractors need to be different between the members of the ensemble. Feature-level combination uses a common classifier that is shared across all members of the ensemble. As such data only needs to be fed through the shared classifier once for the whole ensemble.

An ensemble can also have a diversity of state cluster sets, as is discussed in Section 3.3.3. This form of ensemble diversity relies on each acoustic model discriminating between a different set of state clusters. As such, an ensemble with this form of diversity is only required to have different classifiers, $\mathbf{\Xi}^m$. It is possible to merge the initial feature extractors of the separate acoustic models into a common feature extractor, $\mathbf{\Psi}$, shared across all members of the ensemble [136]. Only separate output layers need to be retained for each set of state clusters. Each output layer with its own set of state clusters represents an individual member

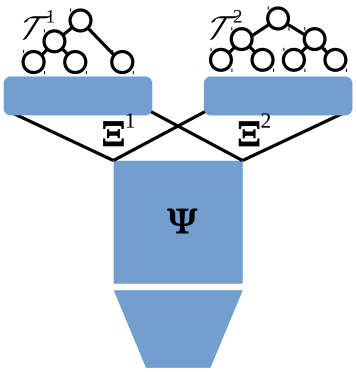


Fig. 4.2 Multi-task ensemble with a separate output layer for each decision tree. Each output layer has parameters Ξ^i, and each set of output nodes are represented by rounded rectangles. Data only needs to be fed through the hidden layers once.

of the ensemble. This topology is referred to as the multi-task ensemble, and is illustrated in Figure 4.2. In this ensemble, data only needs to be fed through the hidden layers once for the whole ensemble. In addition to the computational savings of using this ensemble topology, learning to classify states that have been clustered using multiple different decision trees may also allow the multi-task ensemble to develop a more general hidden representation than when using separate models [7].

All output layers are used when performing recognition, as this is what gives the multi-task ensemble its diversity. This deviates from the multi-task learning methods in [7, 22, 130], which only use the main task when performing recognition. Although the computational cost of performing recognition using a multi-task ensemble is less than when using separate models, it still scales linearly with the ensemble size, $\mathcal{O}(M)$, as data still needs to be fed through each of the output layers. The multi-task ensemble can be combined using either the hypothesis-level combination methods discussed in Section 3.4.1 or the frame-level combination method of (3.40).

One possible method of training the Multi-Task (MT) ensemble is by interpolating together separate cross-entropy criteria for each output layer [136],

$$\mathcal{F}_{\text{MT-CE}}\left(\boldsymbol{\Psi}, \boldsymbol{\Xi}^1, \cdots, \boldsymbol{\Xi}^M\right) = \sum_{m=1}^{M} \lambda_m \sum_{t=1}^{T} \log P\left(s_t^{m,\text{ref}} \middle| \mathbf{o}_t, \boldsymbol{\Psi}, \boldsymbol{\Xi}^m\right), \tag{4.5}$$

where $\boldsymbol{\Xi}^m$ and $s_t^{m,\text{ref}}$ represent the output layer and forced alignment targets respectively for the mth set of state clusters. The interpolation weights satisfy $\lambda_m \geq 0$, and can be used to tune the contribution of each set of state clusters to the ensemble training. The choice of how to obtain the forced alignment state cluster targets for each member of the ensemble may depend on the chosen method of ensemble combination. As is discussed in Section 3.4.2,

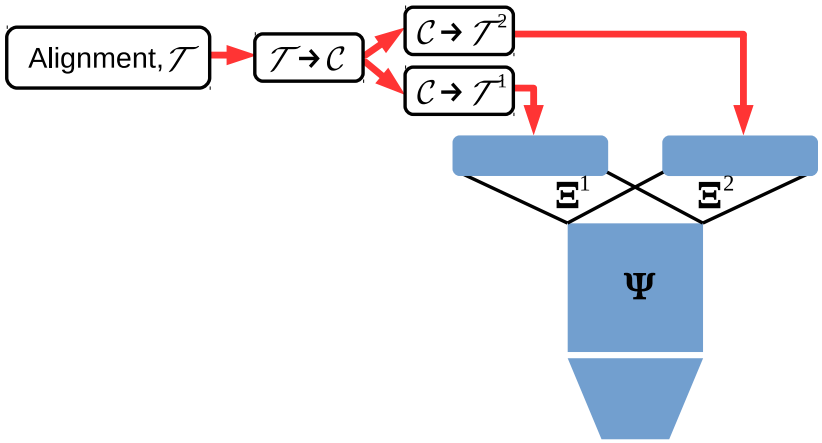


Fig. 4.3 Multi-task cross-entropy training. Forced alignment is mapped from the greedy decision tree, $\mathcal{T}$, to the logical context-dependent states, $\mathcal{C}$, then to each of the random forest trees, $\mathcal{T}^1$ and $\mathcal{T}^2$.

frame-level combination is less computationally expensive, but makes the assumption that all members in the ensemble produce time-synchronous state transitions. This assumption can be incorporated into $\mathcal{F}_{\text{MT-CE}}$ training by using a common set of forced alignments, which are mapped to each of the different state cluster sets, as is illustrated in Figure 4.3.

4.2.1 Joint sequence discriminative training

The cross-entropy criterion trains the multi-task ensemble at the frame-level. However, previous work has shown that performance gains can often be obtained using sequence discriminative training [86]. When using a single system, the $\mathcal{F}_{\text{MBR}}$ criterion of (2.83) can be used to perform sequence discriminative training. It is also possible to train a multi-task ensemble toward a sequence discriminative criterion [134]. One possible method is to interpolation together separate sequence discriminative criteria for each output layer. When using $\mathcal{F}_{\text{MBR}}$ training, this multi-task criteria interpolation can be expressed as

$$\mathcal{F}_{\text{MT-MBR}}\left(\boldsymbol{\Psi}, \boldsymbol{\Xi}^m, \cdots, \boldsymbol{\Xi}^M\right) = \sum_{m=1}^{M} \lambda_m \sum_{\boldsymbol{\omega}} \mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\omega}^{\text{ref}}\right) P\left(\boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Psi}, \boldsymbol{\Xi}^m\right). \tag{4.6}$$

By comparing this form of criterion to MBR combination decoding in (3.29), repeated here for a multi-task ensemble as

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}'} \sum_{m=1}^{M} \lambda_m \sum_{\boldsymbol{\omega}} \mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right) P\left(\boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Psi}, \boldsymbol{\Xi}^m\right), \tag{4.7}$$

it can be seen that this criterion is matched with hypothesis-level combination.

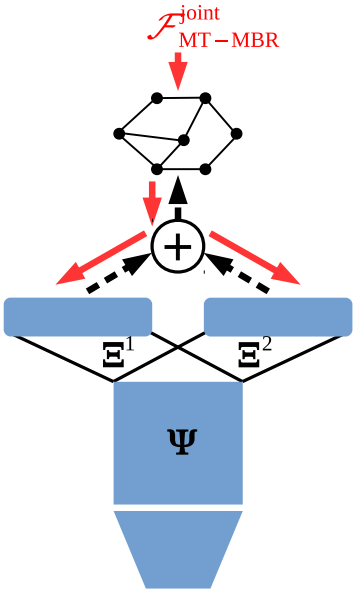


Fig. 4.4 Multi-task joint sequence discriminative training. The criterion derivative is back-propagated through the frame-level combination.

Frame-level combination is less computationally expensive than hypothesis-level combination, as only a single decoding lattice needs to be processed for the whole ensemble. It may be beneficial to train the multi-task ensemble to match the manner in which it is to be combined when performing recognition. Frame-level combination can be performed using (3.40), repeated here for a multi-task ensemble as

$$\mathcal{A}\left(\mathbf{o}_t, c, \boldsymbol{\Psi}, \widehat{\boldsymbol{\Xi}}\right) = \sum_{m=1}^{M} \lambda_m \frac{P\left(s_c^m | \mathbf{o}_t, \boldsymbol{\Psi}, \boldsymbol{\Xi}^m\right)}{P\left(s_c^m\right)}, \tag{4.8}$$

where $\widehat{\boldsymbol{\Xi}} = \{\boldsymbol{\Xi}^m \quad \forall m\}$ represents all of the output layers in the ensemble. These combined scaled observation likelihoods can be used to compute the combined hypothesis posteriors, $P_{\text{frm}}\left(\boldsymbol{\omega}\middle|\mathbf{O}, \boldsymbol{\Psi}, \widehat{\boldsymbol{\Xi}}\right)$, using (2.47). Here, the "frm" subscript is placed to emphasise that these hypothesis posteriors are computed using a frame-level combination.

The multi-task ensemble can be trained by minimising the sequence-level $\mathcal{F}_{\text{MBR}}$ criterion of (2.83), using the frame-level combined hypothesis posteriors,

$$\mathcal{F}^{\text{joint}}_{\text{MT-MBR}}\left(\boldsymbol{\Psi}, \boldsymbol{\Xi}^m, \cdots, \boldsymbol{\Xi}^M\right) = \sum_{\boldsymbol{\omega}} \mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\omega}^*\right) P_{\text{frm}}\left(\boldsymbol{\omega}\middle|\mathbf{O}_{1:T}, \boldsymbol{\Psi}, \widehat{\boldsymbol{\Xi}}\right), \tag{4.9}$$

This method is referred to as joint sequence discriminative training. The derivative can be back-propagated through the frame-level combination to jointly train all of the output layers. This method is published by the author of this thesis in [155]. Using the frame-level

combination during training enforces the time-synchronous state transition requirement that frame-level combination has, as is discussed in Section 3.4.2.

The idea of joint sequence discriminative training of a multi-task ensemble was originally proposed in [134]. However in [134], the chosen frame-level combination method was to take a max over the scaled observation likelihoods,

$$\mathcal{A}\left(\mathbf{o}_t, c, \boldsymbol{\Psi}, \widehat{\boldsymbol{\Xi}}\right) = \max_m \left\{ \frac{P\left(s_c^m | \mathbf{o}_t, \boldsymbol{\Psi}, \boldsymbol{\Xi}^m\right)}{P\left(s_c^m\right)} \right\}, \tag{4.10}$$

which is not differentiable everywhere. Although sub-gradient methods could have been used with such a combination, the work in [134] instead performed sequence discriminative training on a model with a single large output layer, composed of intersect states. As opposed to a max over scaled observation likelihoods, there are many possible frame-level combination methods that are differentiable everywhere, several of which are discussed in [87]. In this thesis, a sum over scaled observation likelihoods of (4.8) is used as the combination method. The derivative through this combination is

$$\frac{\partial \mathcal{A}\left(\mathbf{o}_t, c, \boldsymbol{\Psi}, \widehat{\boldsymbol{\Xi}}\right)}{\partial P\left(s^m | \mathbf{o}_t, \boldsymbol{\Psi}, \boldsymbol{\Xi}^m\right)} = \frac{\lambda_m}{P\left(s_c^m\right)} \delta\left(\mathcal{T}^m\left(c\right), s^m\right). \tag{4.11}$$

It is therefore possible to jointly train all members of the multi-task ensemble toward a sequence discriminative criterion, in a manner that is matched with frame-level combination. Joint sequence discriminative training is also less computationally expensive than $\mathcal{F}_{\text{MT-MBR}}$, as only a single lattice needs to be processed to compute the derivative of the criterion.

4.3 Ensemble compression into a single model

The hypothesis and frame-level combination methods, as well as the multi-task ensemble, all have computational costs that scale as $\mathcal{O}\left(M\right)$ when performing recognition, which scale linearly with the ensemble size. This can be a hindrance when deploying the ASR ensemble on devices with hardware limitations. This section discusses several methods for compressing an ensemble into a single system, such that only one system needs to be used for recognition. This eliminates the linear $\mathcal{O}\left(M\right)$ dependence of the recognition computational cost on the ensemble size. These methods propagate information about the diverse behaviours of the systems within the ensemble into the single compressed system. The nature of the information being propagated affects how much the compressed system is able to learn from the ensemble, and therefore how well it is able to emulate the ensemble performance.

4.3.1 Cross-adaptation

Cross-adaptation is not an ensemble compression method. However, it is a method that propagates information from one system to another, and it is therefore reviewed here for completeness. In cross-adaption, information in the form of 1-best or more hypotheses is propagated from one system to adapt part or the whole of another system. An application where cross-adaptation is commonly used is Speaker Adaptive Training (SAT) using CMLLR [43]. In this implementation of SAT, each speaker has an associated CMLLR linear transformation that is applied to the feature vectors of that speaker. Transcriptions of the utterances are required to train the CMLLR transformations. This presents an issue for unseen speakers in the test set, for whom manual transcriptions are not normally available. Cross-adaptation can be applied to address this issue. Another system, usually a speaker-independent system, can be used to transcribe the utterances of the unseen speakers, and provide 1-best or more hypotheses to train the CMLLR transformations for these speakers. Information about the behaviour of the other transcribing system is propagated to the SAT system and used to learn the CMLLR feature transformations. These CMLLR transformations will therefore capture aspects of the behaviour of the transcribing system. This in effect does a combination of the transcribing system and the SAT system.

Cross-adaptation can also be used to update several or all of the NN layer weights [47]. However, the ability of the resulting adapted system to generalise to unseen data is strongly dependent on the amount of data available to perform cross-adaptation. As such, limiting the number of parameters that are adapted can reduce overfitting when the amount of cross-adaptation data is limited.

4.3.2 Joint ensemble training with diversity-penalisation

The methods discussed in Sections 3.1 and 3.2 generate multiple systems for an ensemble either sequentially or independently. These systems are then combined when performing recognition. One possible combination method is a frame-level combination over state cluster posteriors of (3.33), repeated here as

$$P\left(s\middle|\mathbf{o}_t, \widehat{\mathbf{\Phi}}\right) = \sum_{m=1}^{M} \lambda_m P\left(s|\mathbf{o}_t, \mathbf{\Phi}^m\right). \tag{4.12}$$

This combination method can be computationally expensive, as data needs to be fed through each of the separate acoustic models.

Rather than independently training each member of the ensemble, the work in [171] proposes to jointly train all members of the ensemble, in such a way that each member

behaves similarly to the combined ensemble. In this way, the ensemble behaviour can be approximated by any one of the members of the ensemble, and only a single member of the ensemble needs to be used during recognition. In the work in [171], all of the NN acoustic models are jointly trained, by interpolating together the individual cross-entropy criteria of $\mathcal{F}_{\text{CE}}$ in (2.80), with an additional regularisation term,

$$\mathcal{F}\left(\boldsymbol{\Phi}^1, \cdots, \boldsymbol{\Phi}^M\right) = \sum_{m=1}^{M} \mathcal{F}_{\text{CE}}\left(\boldsymbol{\Phi}^m\right) - \varrho \sum_{m=1}^{M} \sum_{t=1}^{T} \sum_{s \in \mathcal{T}} P\left(s \middle| \mathbf{o}_t, \widehat{\boldsymbol{\Phi}}\right) \log \left[\frac{P\left(s \middle| \mathbf{o}_t, \widehat{\boldsymbol{\Phi}}\right)}{P\left(s | \mathbf{o}_t, \boldsymbol{\Phi}^m\right)} \right], \tag{4.13}$$

where ϱ is the interpolation weight that can be used to tune the contribution of the regularisation term. This regularisation term measures the KL-divergence between the state cluster posteriors of each acoustic model and the frame-level combined posteriors of (4.12). This propagates information about the combined ensemble posteriors to each of the separate models. This may encourage each acoustic model to behave similarly to the combined ensemble.

Although this ensemble generation method results in single systems that emulate the ensemble behaviour, all systems are trained to behave similarly to the combined ensemble. As such, the resulting ensemble may have low diversity and limited combination gains.

4.3.3 Teacher-student learning

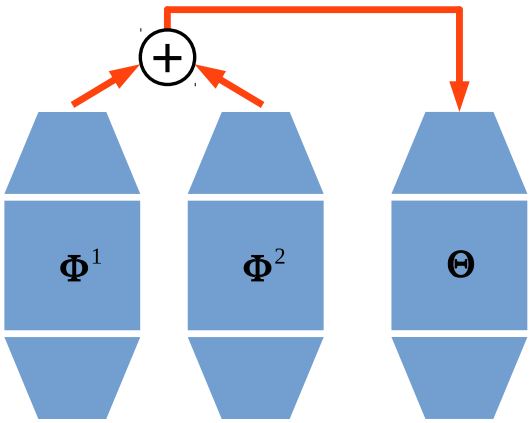


Fig. 4.5 Frame-level teacher-student learning. The per-frame state cluster posteriors of the teachers, $\boldsymbol{\Phi}^1$ and $\boldsymbol{\Phi}^2$, are combined and propagated to the student, $\boldsymbol{\Theta}$.

Rather than propagating information about the combined ensemble to each of its members, this information can be propagated to a different system, so that the members of the ensemble are not constrained to behave similarly. The teacher-student learning method [17] aims to

train a single system, referred to as the student, to emulate the combined behaviour of an ensemble. When performing recognition, only this single student needs to be used. The members in the ensemble are here referred to as the teachers. Teacher-student learning can also be used to train a small student to emulate the behaviour of a single large teacher [93], to also reduce the computational cost when performing recognition.

Teacher-Student (TS) learning aims for the student to emulate the behaviour of, and ideally to produce the same hypotheses as, the combined ensemble. The general approach is to train the student by minimising some distance measure, $\mathbb{D}$, between the behaviours of the teacher ensemble, $\widehat{\mathbf{\Phi}}$ and the student, $\mathbf{\Theta}$,

$$\mathcal{F}\left(\mathbf{\Theta}\right)\big|_{\mathcal{D}} = \mathbb{D}\left\{\widehat{\mathbf{\Phi}}\middle\|\mathbf{\Theta}\right\}\Big|_{\mathcal{D}}. \tag{4.14}$$

This distance is often measured over a training dataset, $\mathcal{D}$. This dependence on the dataset is omitted in the remaining criteria presented in the section, for brevity. The standard method in ASR is to minimise the KL-divergence distance between the frame-level state cluster posteriors of the combined teachers and the student [93],

$$\mathcal{F}_{\text{TS}}^{\text{state}}\left(\mathbf{\Theta}\right) = -\sum_{t=1}^{T}\sum_{s\in\mathcal{T}} P\left(s\middle|\mathbf{o}_t, \widehat{\mathbf{\Phi}}\right)\log P\left(s|\mathbf{o}_t, \mathbf{\Theta}\right). \tag{4.15}$$

In this way, information is propagated from the teachers to the student in the form of frame-level state cluster posteriors.

The targets from the teachers can be combined as a weighted average of the posteriors from each teacher,

$$P\left(s\middle|\mathbf{o}_t, \widehat{\mathbf{\Phi}}\right) = \sum_{m=1}^{M}\lambda_m P\left(s|\mathbf{o}_t, \mathbf{\Phi}^m\right), \tag{4.16}$$

where the interpolation weights satisfy $0 \leq \lambda_m \leq 1$ and $\sum_m \lambda_m = 1$. It is also possible to combine the targets as a product of teacher posteriors, or by using any of the other frame-level combination methods described in [87]. The targets can also be made softer and thus easier to learn by raising them to a power between zero and one [69], $0 \leq \kappa \leq 1$,

$$P'\left(s\middle|\mathbf{o}_t, \widehat{\mathbf{\Phi}}\right) = \frac{P^{\kappa}\left(s\middle|\mathbf{o}_t, \widehat{\mathbf{\Phi}}\right)}{\sum\limits_{s'\in\mathcal{T}} P^{\kappa}\left(s'\middle|\mathbf{o}_t, \widehat{\mathbf{\Phi}}\right)}. \tag{4.17}$$

Here, κ is analogous to being an inverse of the temperature in a statistical physics interpretation. By starting training from a small κ and performing simulated annealing by gradually increasing κ toward 1, the student can initially learn from simpler and higher-entropy tar-

gets, and gradually learn to operate at the entropy level of the original targets. This can be interpreted as a method of obtaining a good initialisation of the student's model parameters. Learning first from high-entropy targets may produce a good initialisation of the student's model parameters, which are then trained toward the lower-entropy targets with larger κ values. This method has been named knowledge distillation [69], because of its similarity to the gradually changing temperature used to separate and propagate liquids from one container to another through distillation.

The standard frame-level teacher-student learning method of $\mathcal{F}_{\text{TS}}^{\text{state}}$ requires that each acoustic model within the teacher ensemble produces state cluster posteriors. The ensemble is therefore allowed to have a diversity of acoustic model parameters and topologies, as well as a diversity of feature representations, as long as all acoustic models produce per-frame posteriors over the same set of state clusters, ensuring that all probability distributions in $\mathcal{F}_{\text{TS}}^{\text{state}}$ have the same support. However, all members in the ensemble are required to use the same set of state clusters, which in turn requires that all members use the same set of sub-word units, HMM topology, and context-dependence. This method also allows for the freedom to choose the student's acoustic model topology and feature representations independently of those used by the teachers in the ensemble. This allows teacher-student learning to be used for compression [93] and domain adaptation [75, 92].

Teacher-student learning trains a full student acoustic model. This is opposed to the common practice in the cross-adaptation method in Section 4.3.1, of only adapting a limit set of the adapted system's parameters. The ability of the student to emulate the ensemble behaviour well on unseen data may be strongly affected by the quantity of data that is available to train the student. However, when training the student, information about the manual transcriptions is not necessary. Therefore it is possible to train the student in a semi-supervised fashion with untranscribed data [93], which is generally cheaper to obtain.

Information in the targets

The standard cross-entropy criterion of (2.80) can be viewed as a specific instance of the teacher-student learning criterion of (4.15), and can be expressed as the minimisation of a KL-divergence,

$$\mathcal{F}_{\text{CE}}\left(\boldsymbol{\Theta}\right) - \sum_{t=1}^{T} \sum_{s \in \mathcal{T}} \delta\left(s, s_t^{\text{ref}}\right) \log P\left(s|\mathbf{o}_t, \boldsymbol{\Theta}\right). \tag{4.18}$$

Here, the targets are the forced alignments, obtained from a system[1], $\boldsymbol{\Phi}$,

$$s_t^{\text{ref}} = \arg\max_s P\left(s_t = s \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}\right). \tag{4.19}$$

[1]This system is often a GMM-HMM, as is discussed in Section 2.3.2.

Because of the similarity between the forms of these two criteria, it is simple to interpolate them together when training the student [69],

$$\mathcal{F}\left(\boldsymbol{\Theta}\right) = \chi \mathcal{F}_{\text{CE}}\left(\boldsymbol{\Theta}\right) + \left(1-\chi\right)\mathcal{F}_{\text{TS}}^{\text{state}}\left(\boldsymbol{\Theta}\right), \tag{4.20}$$

to ground the student to be similar to a cross-entropy-trained system. Here, the interpolation weight satisfies $0 \leq \chi \leq 1$. This is equivalent to interpolating the KL-divergence targets, of the forced alignments and the posteriors produced by the teachers. It is the form of targets that differentiates the cross-entropy and teacher-student learning criteria.

When using teacher-student learning to train a student to emulate an ensemble, it is interesting to question what it is about the targets propagated from the teachers that may benefit the student, above the information that is provided by the forced alignment cross-entropy targets. The form of targets in cross-entropy training only conveys information about which state cluster the teacher, $\boldsymbol{\Phi}$, believes is most likely. Similarly, cross-adaptation can propagate information between systems in the form of 1-best, most likely, hypotheses, as is described in 4.3.1. As opposed to these, the teacher-student targets of $P\left(s\middle|\mathbf{o}_t, \widehat{\boldsymbol{\Phi}}\right)$ may contain information about how difficult the teachers believe each frame is to classify. This is such that frames that are easy to classify can have targets with low entropies, approaching the δ-function cross-entropy targets. On the other hand, frames that are more difficult to classify can have higher entropy targets. Also, the teachers' most likely classes may not agree with each other. Furthermore, there may be frames where the most likely class from the teachers' target differs from the forced alignment target. This may represent frames for which the teachers believe that it may not be possible to classify correctly, or frames for which the forced alignment targets may not represent a time alignment that is the most appropriate for the teachers' model topology. The teacher-student targets therefore in a sense express how uncertain the teachers are about the class of each frame. Assuming that the ensemble of teachers has a greater capacity to model the data than the student, if the teachers are unable to correctly classify a frame, then it may be better for the student to not attempt to produce a low entropy posterior for that frame. Without any information about the difficulty of classifying each frame, in the cross-entropy forced alignment targets, the student must assume that all class labels provided by the forced alignments are absolutely correct, and therefore aims to produce low-entropy posteriors for all frames. This information about how difficult the teachers believe each frame is to classify may therefore benefit the student.

Apart from the standard teacher-student targets of $P\left(s\middle|\mathbf{o}_t, \widehat{\boldsymbol{\Phi}}\right)$, there are in fact many possible forms of targets that can be used to propagate information about how difficult the teachers believe each frame is to classify. A straightforward extension of the cross-entropy criterion to an ensemble of teachers, that captures some degree of the difficulty of classifying

each frame, is to do forced alignment with each teacher separately,

$$s_t^{m,\text{ref}} = \arg\max_s P\left(s_t = s \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right), \tag{4.21}$$

and then to take a linear combination of these 1-best forced alignment targets,

$$\mathcal{F}_{\text{TS}}^{\text{1best}}(\boldsymbol{\Theta}) = -\sum_{t=1}^{T}\sum_{s\in\mathcal{T}}\left[\sum_{m=1}^{M}\lambda_m\delta\left(s, s_t^{m,\text{ref}}\right)\right]\log P\left(s|\mathbf{o}_t, \boldsymbol{\Theta}\right). \tag{4.22}$$

The forced alignments from each teacher may differ in their pronunciation variants and their state transition times. These differences may occur at frames with a high ambiguity as to which state cluster it belongs to. Such information may be useful when training a student. However, using only the 1-best alignments from each teacher only captures a limited amount of information about the disagreement between the teachers. It also does not express any information about how certain each individual teacher is about the class of each frame. Furthermore, unlike the teacher targets here, the student has no access to information about the manual transcriptions when producing its state cluster posteriors. Without this information, there may be frames where the student is not able to effectively emulate this form of targets.

Information about how certain each teacher is about the class of each frame can be propagated by using targets that consider the full alignment distribution from each teacher, instead of just the 1-best alignments,

$$\mathcal{F}_{\text{TS}}^{\text{soft-align}}(\boldsymbol{\Theta}) = -\sum_{t=1}^{T}\sum_{s\in\mathcal{T}}\left[\sum_{m=1}^{M}\lambda_m P\left(s_t = s \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right)\right]\log P\left(s|\mathbf{o}_t, \boldsymbol{\Theta}\right). \tag{4.23}$$

The derivative of this criterion is

$$\frac{\partial\mathcal{F}_{\text{TS}}^{\text{soft-align}}(\boldsymbol{\Theta})}{\partial z_{st}^{(K+1)}} = P\left(s|\mathbf{o}_t, \boldsymbol{\Theta}\right) - \sum_{m=1}^{M}\lambda_m P\left(s_t = s \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right). \tag{4.24}$$

This is similar to the $\mathcal{F}_{\text{CTC}}$ derivative of (2.131), if the target soft alignments are obtained from the student, $\boldsymbol{\Theta}$, instead of the teachers, $\boldsymbol{\Phi}^m$. Similarly to the $\mathcal{F}_{\text{CTC}}$ criterion in (2.130), the targets here consider all possible alignments of the state sequences that can represent the manual transcriptions, and not just the 1-best alignment. This may be able to capture the uncertainty that each teacher has about which pronunciation variants and state transition times are the most appropriate, in addition to the disagreements between the teachers. However, the alignment distributions in the targets of $\mathcal{F}_{\text{TS}}^{\text{soft-align}}$ again utilise information about the manual

transcriptions, which the student has no access to when producing its state cluster posteriors. Teacher-student learning using the $\mathcal{F}_{\text{TS}}^{\text{soft-align}}$ criterion is investigated in [71].

Using targets combined with (4.16), the standard teacher-student criterion of $\mathcal{F}_{\text{TS}}^{\text{state}}$ in (4.15) can be expressed as

$$\mathcal{F}_{\text{TS}}^{\text{state}}\left(\boldsymbol{\Theta}\right) = -\sum_{t=1}^{T}\sum_{s\in\mathcal{T}}\left[\sum_{m=1}^{M}\lambda_m P\left(s|\mathbf{o}_t,\boldsymbol{\Phi}^m\right)\right]\log P\left(s|\mathbf{o}_t,\boldsymbol{\Theta}\right). \tag{4.25}$$

Here, the teacher targets and student posteriors are conditioned on the same variables. As such, the difficulty of classifying each frame, expressed in teacher targets may be better matched with the posteriors that the student can produce.

4.3.4 Parameter-level combination

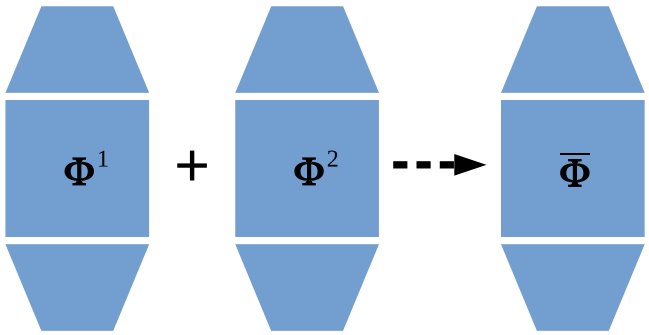


Fig. 4.6 Parameter-level combination takes an average of the model parameters.

The combination methods discussed in Section 3.4 combine together either hypothesis or frame-level probabilities, or features. Another possible method is to directly combine the model parameters. As with the other compression methods discussed in Section 4.3, this also results in there only being a single system that needs to be used when performing recognition. This parameter-level combination can be achieved as a weighted average of the model parameters [141],

$$\overline{\boldsymbol{\Phi}} = \sum_{m=1}^{M}\lambda_m\boldsymbol{\Phi}^m. \tag{4.26}$$

The resulting single model, $\overline{\boldsymbol{\Phi}}$, is referred to as the smoothed model. This combination method is not probabilistic in nature. As such, the interpolation weights, λ_m do not strictly need to be constrained to be non-negative or sum to one. These interpolation weights can be set manually or optimised toward a chosen training criterion [114]. If the weights are set manually, then unlike the teacher-student learning and cross-adaptation methods discussed

in Sections 4.3.3 and 4.3.1, no training is required to construct the smoothed model. In teacher-student learning and cross-adaptation, the ability of the student or adapted system to generalise well to unseen data depends on the quantity of training data that is available.

The combination of (4.26) only uses a single interpolation weight for each member of the ensemble. The combination can be generalised to allow for a separate weight for each model parameter or group of model parameters,

$$\overline{\mathbf{\Phi}} = \sum_{m=1}^{M} \boldsymbol{\lambda}_m \odot \mathbf{\Phi}^m, \tag{4.27}$$

where $\odot$ represents element-wise multiplication. However, this may greatly increase the number of combination parameters that need to be set. If these weights are set by optimising them toward a chosen criterion, then it may be difficult to train them to generalise well, with a limited quantity of training data.

Parameter-level combination using (4.26) requires that all models have the same topology. It is possible to permute the NN parameters without affecting the resulting behaviour of the model. However, the permuted NN will have hidden units that are re-ordered. The models combined with parameter-level combination must have hidden units that are ordered similarly. These constraints may limit the diversity of an ensemble that can be used with this combination method. One possible method of abiding by these constraints is to construct an ensemble out of models at intermediate iterations of a single run of training [23]. This is described in Section 3.2.4.

Another possible method to generate an ensemble that abides by these constraints is by using Dropout. Section 3.1.3 describes how Dropout can be used to generate an ensemble. After a single model has been trained, multiple models can be obtained when performing recognition, by feeding data through the single model multiple times, each time with a different Dropout mask sample. Each Dropout mask sample, $\mathbf{d}^{(k)m}$, effectively produces a new model, $\mathbf{\Phi}^m = \left\{\mathbf{W}^{(k)m}, \mathbf{b}^{(k)m} \quad \forall k\right\}$, according to (3.14), which is repeated here,

$$\mathbf{W}^{(k)m} = \frac{1}{1-\pi}\mathbf{W}^{(k)}\mathbf{Diag}\left(\mathbf{d}^{(k)m}\right) \tag{4.28}$$

$$\mathbf{b}^{(k)m} = \mathbf{b}^{(k)}. \tag{4.29}$$

These models are sufficiently similar to each other, such that they can be used with parameter-level combination. Parameter-level combination of this Dropout ensemble can be achieved simply by using the model without Dropout, $\breve{\mathbf{\Phi}} = \left\{\mathbf{W}^{(k)}, \mathbf{b}^{(k)} \quad \forall k\right\}$, when performing recognition. The Dropout mask is sampled from a Bernoulli distribution in

(2.109), repeated here as

$$P\left(d_i^{(k)m}\middle|\pi\right) = \pi^{1-d_i^{(k)m}}\left(1-\pi\right)^{d_i^{(k)m}}. \tag{4.30}$$

The expected value of each mask unit is

$$\mathbb{E}\left\{d_i^{(k)m}\right\} = \lim_{M\to\infty}\sum_{m=1}^{M}\frac{1}{M}d_i^{(k)m} \tag{4.31}$$

$$= 1-\pi. \tag{4.32}$$

Taking an equally weighted parameter-level combination of an infinite number of models in a Dropout ensemble results in

$$\overline{\mathbf{W}^{(k)}} = \lim_{M\to\infty}\sum_{m=1}^{M}\frac{1}{M\left(1-\pi\right)}\mathbf{W}^{(k)}\mathbf{Diag}\left(\mathbf{d}^{(k)m}\right) \tag{4.33}$$

$$= \mathbf{W}^{(k)}. \tag{4.34}$$

Therefore the parameter-level combined Dropout ensemble is equivalent to the model that is used without Dropout, $\overline{\mathbf{\Phi}} = \breve{\mathbf{\Phi}}$. As with an ensemble generated from the intermediate model iterations in a single run of training, a Dropout ensemble also only requires a single training run.

In [137], training a model with Dropout, and then using it without Dropout when performing recognition is interpreted as a method of regularisation. This same method can equivalently be viewed as a parameter-level combination of an ensemble. This relationship may suggest that there is some equivalence between using an ensemble and performing regularisation.

4.4 Summary

This chapter has considered various schemes for reducing the computational cost of performing recognition using an ensemble. The number of parameters within each system can be reduced using low-rank matrix factorisation. When using a diversity of state cluster sets, an ensemble can be compressed by tying together the hidden layer parameters, leaving only separate output layers for each set of state clusters. This multi-task ensemble topology only requires data to be fed through the hidden layers once for the whole ensemble. An ensemble can be further compressed into a single system. Teacher-student learning achieves this by training a student to emulate the combined ensemble behaviour. Different forms of

information, that capture how difficult the ensemble believes that each frame is to classify, can be propagated to the student. Parameter-level combination can also be used to compress an ensemble into a single system. However, this requires all acoustic models to have the same topology and similarly ordered hidden representations, which may limit the ensemble diversity.

Chapter 5

Frame-level teacher-student learning with diverse teachers

Teacher-student learning, described in Section 4.3.3 can be used to compress an ensemble into a single student, thereby reducing the computational cost when performing recognition. The standard teacher-student learning method propagates frame-level state cluster posterior information from the teachers to the student. This requires that the teacher and student acoustic models must produce state cluster posteriors over the same set of state clusters. This constraint limits the forms of diversities that are allowed between the teachers in the ensemble.

This chapter proposes to generalise frame-level teacher-student learning to allow for more diverse teachers. First, Section 5.1 investigates methods of compressing an ensemble that uses a diversity of state cluster sets. Two methods are proposed, that either map the teachers' state cluster posteriors to the student's set of state clusters, or train a multi-task ensemble, discussed in Section 4.2, to emulate the ensemble of separate systems. Second, Section 5.2 considers how frame-level teacher-student learning can be used with an ensemble whose systems use the lattice-free topology, described in Section 2.5.2. In this topology, the acoustic model directly produces log-acoustic scores at its output, as in (2.119), instead of state cluster posteriors. Therefore, state cluster posteriors cannot be obtained directly from these systems. Frame-level teacher-student learning needs to be modified to be used with such systems.

5.1 Learning from different sets of state clusters

As has been discussed in Section 3.3.3, an ensemble can incorporate a diversity of state cluster sets. Using different sets of state clusters may allow the members of the ensemble to exhibit diverse behaviours. However, as with many other forms of ensembles, the computational cost of performing recognition can be high. Teacher-student learning, discussed in Section 4.3.3, is one possible method of compressing an ensemble and reducing the computational cost of performing recognition. However, the standard teacher-student learning method in (4.15) requires the state cluster posteriors from all acoustic models to have the same support. This requires all members of the ensemble to use the same set of state clusters, and may therefore limit the ensemble diversity.

5.1.1 Mapping posteriors across state cluster sets

Standard teacher-student learning in (4.15) minimises a KL-divergence between state cluster posteriors. This thesis proposes to allow the ensemble to have a diversity of state cluster sets, by modifying the teacher-student learning criterion to instead minimise a KL-divergence over a set of acoustic units that are common across all systems. One possible choice of such an acoustic unit are the logical Context-Dependent (CD) states, c, leading to a criterion of

$$\mathcal{F}_{\text{TS}}^{\text{CD}}\left(\boldsymbol{\Theta}\right) = -\sum_{t=1}^{T}\sum_{c\in\mathcal{C}} P\left(c\middle|\mathbf{o}_t, \widehat{\boldsymbol{\Phi}}\right)\log P\left(c|\mathbf{o}_t, \boldsymbol{\Theta}\right), \tag{5.1}$$

where $\mathcal{C}$ is the set of all logical context-dependent states. This is proposed by the author of this thesis in [156]. The criterion requires the posteriors of the logical context-dependent states, $P\left(c|\mathbf{o}_t, \boldsymbol{\Theta}\right)$. Since there is a unique mapping from c to s_c by using (2.28), the logical context-dependent state posteriors can be expressed as

$$\begin{aligned} P\left(c|\mathbf{o}_t, \boldsymbol{\Theta}\right) &= \sum_{s^{\boldsymbol{\Theta}}\in\mathcal{T}^{\boldsymbol{\Theta}}} P\left(c\middle|s^{\boldsymbol{\Theta}}, \mathbf{o}_t, \boldsymbol{\Theta}'\right) P\left(s^{\boldsymbol{\Theta}}\middle|\mathbf{o}_t, \boldsymbol{\Theta}\right) &(5.2)\\ &= P\left(c\middle|s_c^{\boldsymbol{\Theta}}, \mathbf{o}_t, \boldsymbol{\Theta}'\right) P\left(s_c^{\boldsymbol{\Theta}}\middle|\mathbf{o}_t, \boldsymbol{\Theta}\right), &(5.3)\end{aligned}$$

where $\sum_{s^{\boldsymbol{\Theta}}\in\mathcal{T}^{\boldsymbol{\Theta}}}$ sums over all state clusters, $s^{\boldsymbol{\Theta}}$, at the leaves of the student's decision tree, $\mathcal{T}^{\boldsymbol{\Theta}}$. This allows the criterion to be expressed as

$$\mathcal{F}_{\text{TS}}^{\text{CD}}\left(\boldsymbol{\Theta}\right) = -\sum_{t=1}^{T}\sum_{c\in\mathcal{C}} P\left(c\middle|\mathbf{o}_t, \widehat{\boldsymbol{\Phi}}\right)\left[\log P\left(s_c^{\boldsymbol{\Theta}}\middle|\mathbf{o}_t, \boldsymbol{\Theta}\right) + \log P\left(c\middle|s_c^{\boldsymbol{\Theta}}, \mathbf{o}_t, \boldsymbol{\Theta}'\right)\right]. \tag{5.4}$$

The standard hybrid NN-HMM ASR system does not capture $P\left(c\middle|s_c^{\boldsymbol{\Theta}}, \mathbf{o}_t, \boldsymbol{\Theta}'\right)$. Capturing this would allow the observation likelihood of each logical context-dependent state to be separately computed, going against the motivation of state tying with a decision tree, to reduce the number of model parameters, discussed in 2.2.4. One possible solution around this limitation is to assume that this distribution is independent of the acoustic model parameters, $\boldsymbol{\Theta}$. With this independence assumption, $P\left(c\middle|s_c^{\boldsymbol{\Theta}}, \mathbf{o}_t, \boldsymbol{\Theta}'\right)$ can be ignored when computing the gradient to train $\boldsymbol{\Theta}$.

The criterion can be simplified to consider a sum over state clusters in the student's decision tree, rather than over logical context-dependent states,

$$\mathcal{F}_{\text{TS}}^{\text{CD}'}\left(\boldsymbol{\Theta}\right) = -\sum_{t=1}^{T}\sum_{s^{\boldsymbol{\Theta}}\in\mathcal{T}^{\boldsymbol{\Theta}}} P\left(s^{\boldsymbol{\Theta}}\middle|\mathbf{o}_t, \widehat{\boldsymbol{\Phi}}\right)\log P\left(s^{\boldsymbol{\Theta}}\middle|\mathbf{o}_t, \boldsymbol{\Theta}\right). \tag{5.5}$$

The student can be trained using this criterion, as long as the targets of $P\left(s^{\boldsymbol{\Theta}}\middle|\mathbf{o}_t, \widehat{\boldsymbol{\Phi}}\right)$ can be obtained. These targets represent posteriors of the student's set of state clusters, obtained from the teachers. The challenge for performing teacher-student learning is to compute these targets.

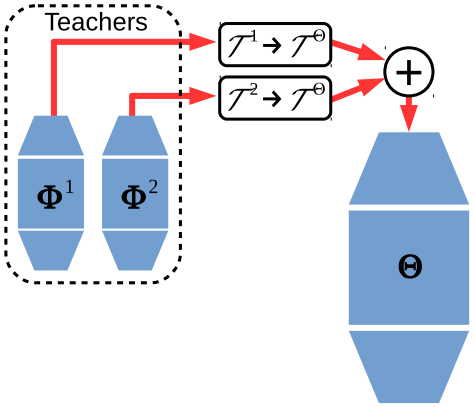


Fig. 5.1 Frame-level teacher-student learning across different sets of state clusters. Per-frame posteriors from the teachers are mapped from the teachers' decision trees, $\mathcal{T}^1$ and $\mathcal{T}^2$, to the student's decision tree, $\mathcal{T}^{\boldsymbol{\Theta}}$.

The teachers' posteriors can be combined using any of the frame-level methods discussed in Section 3.4.2. When using a sum combination, similarly to (4.16), the targets can be expressed as

$$P\left(s^{\boldsymbol{\Theta}}\middle|\mathbf{o}_t, \widehat{\boldsymbol{\Phi}}\right) = \sum_{m=1}^{M}\lambda_m P\left(s^{\boldsymbol{\Theta}}\middle|\mathbf{o}_t, \boldsymbol{\Phi}^m\right). \tag{5.6}$$

By using (5.2), but for the teachers, the posteriors of each of the teachers are mapped to the student's set of state clusters as

$$P\left(s^{\mathbf{\Theta}}\middle|\mathbf{o}_t,\mathbf{\Phi}^m\right) = \sum_{c:\mathcal{T}^{\mathbf{\Theta}}(c)=s^{\mathbf{\Theta}}} P\left(c|\mathbf{o}_t,\mathbf{\Phi}^m\right) \tag{5.7}$$

$$= \sum_{c:\mathcal{T}^{\mathbf{\Theta}}(c)=s^{\mathbf{\Theta}}} \sum_{s^m\in\mathcal{T}^m} P\left(c|s^m,\mathbf{o}_t,\mathbf{\Phi}^{m\prime}\right) P\left(s^m|\mathbf{o}_t,\mathbf{\Phi}^m\right). \tag{5.8}$$

This can be expressed as

$$P\left(s^{\mathbf{\Theta}}\middle|\mathbf{o}_t,\mathbf{\Phi}^m\right) = \sum_{s^m\in\mathcal{T}^m} P\left(s^{\mathbf{\Theta}}\middle|s^m,\mathbf{o}_t,\mathbf{\Phi}^{m\prime}\right) P\left(s^m|\mathbf{o}_t,\mathbf{\Phi}^m\right), \tag{5.9}$$

where

$$P\left(s^{\mathbf{\Theta}}\middle|s^m,\mathbf{o}_t,\mathbf{\Phi}^{m\prime}\right) = \sum_{c:\mathcal{T}^{\mathbf{\Theta}}(c)=s^{\mathbf{\Theta}}} P\left(c|s^m,\mathbf{o}_t,\mathbf{\Phi}^{m\prime}\right) \tag{5.10}$$

represents a mapping from the posteriors over the teachers' sets of state clusters, $P\left(s^m|\mathbf{o}_t,\mathbf{\Phi}^m\right)$, to the student's set of state clusters, $P\left(s^{\mathbf{\Theta}}\middle|\mathbf{o}_t,\mathbf{\Phi}^m\right)$, illustrated in Figure 5.1. If the same set of state clusters is used for all systems, then $P\left(s^{\mathbf{\Theta}}\middle|s^m,\mathbf{o}_t,\mathbf{\Phi}^{m\prime}\right) = \delta\left(s^{\mathbf{\Theta}},s^m\right)$ becomes a δ-function, and $\mathcal{F}_{\text{TS}}^{\text{CD}\prime}$ in (5.5) reduces back to the standard teacher-student learning criterion in (4.15).

The computation of the posterior map, $P\left(s^{\mathbf{\Theta}}\middle|s^m,\mathbf{o}_t,\mathbf{\Phi}^{m\prime}\right)$, requires the computation of the posteriors of each of the individual logical context-dependent states, using $P\left(c|s^m,\mathbf{o}_t,\mathbf{\Phi}^{m\prime}\right)$. However, as was previously mentioned, because of state tying, these are not individually separable for each of the logical context-dependent states. One possible solution to obtain the map is to make the approximation that it is independent of the observations,

$$P\left(s^{\mathbf{\Theta}}\middle|s^m,\mathbf{o}_t,\mathbf{\Phi}^{m\prime}\right) \approx P\left(s^{\mathbf{\Theta}}\middle|s^m\right) \tag{5.11}$$

$$= \sum_{c:\mathcal{T}^{\mathbf{\Theta}}(c)=s^{\mathbf{\Theta}}} \frac{P\left(c\right)}{\sum\limits_{c':\mathcal{T}^m(c')=s^m} P\left(c'\right)} \delta\left(\mathcal{T}^m\left(c\right),s^m\right). \tag{5.12}$$

The logical context-dependent state priors, $P\left(c\right)$, can be estimated using a discounted maximum likelihood estimate from the forced alignments,

$$P\left(c\right) = \frac{N_c+\nu}{\sum\limits_{c'\in\mathcal{C}}\left(N_{c'}+\nu\right)}, \tag{5.13}$$

where N_c is the number of times that logical context-dependent state c appears in the forced alignments. The set of logical context-dependent states, $\mathcal{C}$, may be large, and not all logical

context-dependent states may occur within a finite amount of training data. Discounting with ν can improve the generalisation of the approximate map, for logical context-dependent states that are not seen in the training data.

Using the approximation of (5.11) allows the contribution to the targets from each teacher in (5.9) to be expressed as

$$P\left(s^{\boldsymbol{\Theta}}\middle|\mathbf{o}_t, \boldsymbol{\Phi}^m\right) = \sum_{s^m \in \mathcal{T}^m} P\left(s^{\boldsymbol{\Theta}}\middle|s^m\right) P\left(s^m|\mathbf{o}_t, \boldsymbol{\Phi}^m\right). \tag{5.14}$$

Here, the approximate map of $P\left(s^{\boldsymbol{\Theta}}\middle|s^m\right)$ is used to transform the posteriors from the teachers' sets of state clusters to the student's set of state clusters. Using this, targets can be obtained to train the student with the $\mathcal{F}_{\text{TS}}^{\text{CD}\prime}$ criterion in (5.5), and frame-level teacher-student learning can be used when the sets of state clusters differ between the teachers and the student. Information about the state cluster posteriors for each different set of state clusters undergoes this mapping, when being propagated from the teachers to the student. However, the approximation of (5.11), made to obtain this map, may result in a loss of propagated information.

The proposed criterion of $\mathcal{F}_{\text{TS}}^{\text{CD}\prime}$ allows an ensemble with a diversity of state cluster sets to be compressed into a single student. Only the student needs to be used when performing recognition. However, the student only models the state clusters defined by a single decision tree, while the members in the ensemble model multiple sets of state clusters. As is discussed in Section 3.4.2, the maximum phonetic resolution that the ensemble can have is represented by the intersect states, formed by a Cartesian product of all of the decision trees in the ensemble. The set of intersect states may be large. As such, a student may need to use a large set of state clusters in order to adequately emulate the ensemble behaviour. The proposed criterion allows for the freedom to choose the student's set of state clusters independently of the sets of state clusters that the teachers in the ensemble use. However, using a larger set of state clusters requires a larger output layer, which incurs a greater computational cost when performing recognition and may be more difficult to train to generalise well. The freedom of choice of the student's set of state clusters, provided by the proposed criterion, can be used to find an acceptable balance between the computational cost of performing recognition and the output complexity of the student's acoustic model.

5.1.2 Multi-task teacher-student learning

Compressing an ensemble with a diversity of state cluster sets into a student may result in losses due to the approximate posterior map of (5.11) and the student may need to use a large

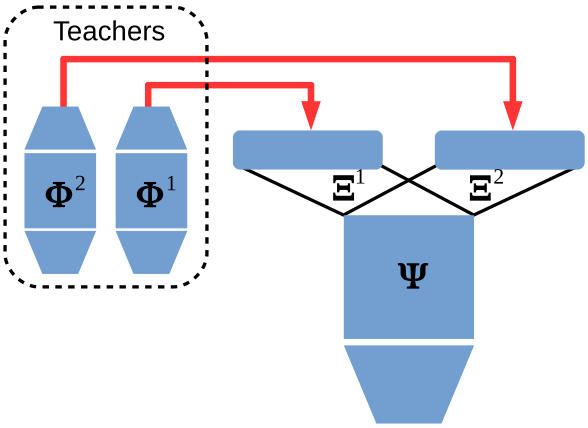


Fig. 5.2 Multi-task teacher-student learning. Targets are obtained from separate teachers, rather than from forced alignment. There is no need to map the teachers' posteriors between difference decision trees.

set of state clusters to effectively emulate the ensemble behaviour. As is discussed in Section 4.2, an alternative method of compressing an ensemble with a diversity of state cluster sets is to use a multi-task ensemble. Rather than using a student with a single output layer, an ensemble of separate models can instead be compressed into a multi-task ensemble, by using teacher-student learning to train the multi-task ensemble to emulate the behaviours of the separate models.

This thesis proposes to train the multi-task ensemble by minimising the KL-divergence between the per-frame state cluster posteriors of each state cluster set, between the separate models as the teachers and the multi-task ensemble as the student,

$$\mathcal{F}_{\text{MT-TS}}\left(\mathbf{\Psi}, \mathbf{\Xi}^1, \cdots, \mathbf{\Xi}^M\right) = -\sum_{m=1}^{M} \lambda_m \sum_{t=1}^{T} \sum_{s^m \in \mathcal{T}^m} P\left(s^m | \mathbf{o}_t, \mathbf{\Phi}^m\right) \log P\left(s^m | \mathbf{o}_t, \mathbf{\Psi}, \mathbf{\Xi}^m\right). \tag{5.15}$$

Here again, $\mathbf{\Psi}$ are the hidden layer parameters that are shared across all members of the multi-task ensemble, and $\mathbf{\Xi}^m$ are the separate output layer parameters for each multi-task ensemble member. The interpolation weights satisfy $\lambda_m \geq 0$ and can be used to tune the contribution of each set of state clusters during training. This method is proposed by the author of this thesis in [155], and is illustrated in Figure 5.2. It may be more advantageous to use a multi-task ensemble than a student with a single output layer, as the multi-task ensemble can use the same sets of state clusters as the separate teachers, and therefore can express the same phonetic resolution as the combined separate models. Furthermore, no approximate map of the form of (5.11) is required to obtain the targets needed to train the multi-task ensemble.

However, when using a multi-task ensemble, data needs to be fed through each of the separate output layers when performing recognition. Therefore, using the multi-task ensemble to perform recognition has a computational cost that scales as $\mathcal{O}(M)$, increasing linearly with the number of different sets of state clusters, M. On the other hand, the computational cost of using a student with a single output layer to perform recognition depends on the size of the student's set of state clusters. If the student uses a large set of state clusters, approaching the number of intersect states of the teacher ensemble, then it may be more computationally expensive to use this student for recognition than to use a multi-task ensemble.

Multi-task teacher-student learning can be compared with an alternative method of training the multi-task ensemble, using the mutli-task cross-entropy criterion in (4.5), discussed in Section 4.2. This criterion can be re-expressed as a minimisation of KL-divergences,

$$\mathcal{F}_{\text{MT-CE}}\left(\boldsymbol{\Psi}, \boldsymbol{\Xi}^1, \cdots, \boldsymbol{\Xi}^M\right) = -\sum_{m=1}^{M} \lambda_m \sum_{t=1}^{T} \sum_{s^m \in \mathcal{T}^m} \delta\left(s^m, s^{m,\text{ref}}\right) \log P\left(s^m | \mathbf{o}_t, \boldsymbol{\Psi}, \boldsymbol{\Xi}^m\right), \tag{5.16}$$

where the targets of $\delta\left(s^m, s^{m,\text{ref}}\right)$ are obtained from forced alignments. By training the multi-task ensemble using the multi-task cross-entropy criterion, the multi-task ensemble learns to develop diverse behaviours, as each member aims to discriminate between a different set of state clusters. However, the multi-task ensemble has its hidden layer parameters tied across all members of the ensemble. This leads to a reduction in the number of parameters, compared to an ensemble of separate models. This may result in a loss in the diversity between the behaviours of the multi-task ensemble members. Using teacher-student learning, frame-level state cluster posterior information from the separate models is propagated to the multi-task ensemble in the targets of $P\left(s^m | \mathbf{o}_t, \boldsymbol{\Phi}^m\right)$. This information may allow the multi-task ensemble to learn from the diverse behaviours of the separate models.

5.2 Learning from lattice-free systems

Section 2.5.2 has discussed the lattice-free sequence discriminative training method. In this method, there is no need to pre-compute and store pruned lattices during training. Instead, a simplified recognition graph is used to generate a lattice for each utterance on-the-fly, which contains all hypotheses allowed by the graph. Since there is no need of performing lattice pruning, there is in turn no need of an initial acoustic model to provide acoustic scores. In the lattice-based method, the initial acoustic model used to provide reasonable acoustic scores for pruning is often trained using the cross-entropy criterion. When an ensemble is constructed out of lattice-based systems, the systems may share a common bias toward the cross-entropy forced alignments. Since there is no need for initial acoustic scores in the

lattice-free method, it is possible to begin sequence discriminative training from a random parameter initialisation. An ensemble of lattice-free systems will therefore not be commonly biased toward cross-entropy forced alignments. This may lead to a wider diversity between the behaviours of lattice-free systems.

An ensemble of lattice-free systems may be able to leverage upon this wider diversity. However, the ensemble can again be computationally expensive to use for recognition. Teacher-student learning can be used to compress the ensemble into a single student. The standard frame-level teacher-student learning criterion in (4.15) requires frame-level state cluster posteriors to be obtained from each of the acoustic models. However, lattice-free acoustic models are often designed to directly produce log-acoustic scores at their outputs using (2.119), rather than state cluster posteriors, as is described in Section 2.5.2. This is because there is no need for an initial acoustic model to produce reasonable acoustic scores to prune the lattice, and therefore there is no need to perform cross-entropy training to obtain this initial model. It is the initial cross-entropy training that requires the acoustic model to produce state cluster posteriors for the criterion computation of (2.80) and the derivative computation of (2.110). As such, it may not be trivial to obtain state cluster posteriors from lattice-free systems. Therefore, frame-level teacher-student learning needs to be modified to be used with the lattice-free acoustic model topology.

As is described in Section 4.3.3, the aim of teacher-student learning is to train a student to emulate the behaviour of the teachers. A general approach to this is to minimise some distance measure between the behaviours of the teachers and student, described by (4.14). Since state cluster posteriors are not readily obtainable when using the Lattice-Free (LF) acoustic model topology, one possible alternative criterion is to minimise the Mean Squared Error (MSE) between the log-acoustic scores of the combined ensemble and the student, produced at the linear outputs of the acoustic models using (2.119) [79],

$$\mathcal{F}^{\text{MSE}}_{\text{LF-TS}}\left(\boldsymbol{\Theta}\right) = \frac{1}{2}\sum_{t=1}^{T}\sum_{s\in\mathcal{T}}\left[\sum_{m=1}^{M}\lambda_m \log\mathcal{A}\left(\mathbf{o}_t, s, \boldsymbol{\Phi}^m\right) - \log\mathcal{A}\left(\mathbf{o}_t, s, \boldsymbol{\Theta}\right)\right]^2, \tag{5.17}$$

where the interpolation weights satisfy $\lambda_m \geq 0$ and can be used to tune the contribution of each teacher during training.

This thesis proposes another criterion that may be used to train the student with a lattice-free acoustic model topology. When computing the hypothesis posteriors of a lattice-free system in (2.120), the acoustic scores, $\mathcal{A}\left(\mathbf{o}_t, s, \boldsymbol{\Theta}\right)$, are computed by taking the exponential of the NN acoustic model output. Therefore, the acoustic scores are always positive. As such, it is possible to convert these into something that looks like a probability distribution through

normalisation. The KL-divergence distance measure can then be used as

$$\widetilde{\mathcal{F}}^{\text{KL}}_{\text{LF-TS}}\left(\boldsymbol{\Theta}\right) = -\sum_{t=1}^{T}\sum_{s\in\mathcal{T}}\sum_{m=1}^{M}\lambda_m \frac{\mathcal{A}\left(\mathbf{o}_t, s, \boldsymbol{\Phi}^m\right)}{\sum\limits_{s'\in\mathcal{T}}\mathcal{A}\left(\mathbf{o}_t, s', \boldsymbol{\Phi}^m\right)} \log \frac{\mathcal{A}\left(\mathbf{o}_t, s, \boldsymbol{\Theta}\right)}{\sum\limits_{s'\in\mathcal{T}}\mathcal{A}\left(\mathbf{o}_t, s', \boldsymbol{\Theta}\right)}, \tag{5.18}$$

where the interpolation weights satisfy $0 \leq \lambda_m \leq 1$ and $\sum_m \lambda_m = 1$. This criterion allows the existing frame-level teacher-student learning infrastructure to be re-used. Although the teachers in (5.18) are combined as a sum combination, any other frame-level combination method is also possible. When using a sum combination, it may be important to perform normalisation for each teacher separately before combination, rather than combining the acoustic scores and normalising the combined score. This is because the linear outputs of the NNs that form the log-acoustic scores in (2.119) are unbounded, and may result in there being significantly different dynamic ranges between the acoustic scores of each of the teachers. Separately normalising the contributions from each teacher may help to prevent any single teacher from dominating the combination.

The unbounded nature of the linear outputs of the lattice-free model topology may potentially present numerical stability issues when used with frame-level teacher-student learning. The linear NN outputs are interpreted as log-acoustic scores, expressed as (2.119). These are converted to acoustic scores by taking the exponential, and then used to compute the hypothesis posteriors using (2.120). Adding an equal constant offset to all output nodes will not affect the resulting hypothesis posterior or the computed criterion of $\widetilde{\mathcal{F}}^{\text{KL}}_{\text{LF-TS}}$. However, a large positive linear NN output value may result in numerical overflow when taking its exponential to compute the acoustic score, if no safeguards are implemented. In lattice-free $\mathcal{F}_{\text{MMI}}$ training in [116], the linear NN outputs can be prevented from growing too large by minimising the L2 norm of these linear NN outputs as an additional regularisation term. It is possible to incorporate the same regularisation term into frame-level teacher-student learning. A simpler solution, adopted in the experiments in this thesis, is to train the student without the regularisation term. After the student has been trained with $\widetilde{\mathcal{F}}^{\text{KL}}_{\text{LF-TS}}$, a constant offset can be subtracted equally from all of the student's linear NN output nodes. This constant offset can be computed as the expected L2 norm of the student's linear NN outputs, averaged across all output nodes.

5.3 Summary

This chapter has considered generalisations of frame-level teacher-student learning to allow it to be used to compress more diverse forms of ensembles. A diversity of state cluster sets is

allowed by minimising the KL-divergence between per-frame posteriors of logical context-dependent states, instead of state clusters. This leads to a method of mapping the posteriors from the teachers' sets of state clusters to that of the student. However, this mapping requires approximation and the student may need to use a large set of state clusters to effectively capture the same phonetic resolution as the ensemble. Instead, the ensemble of separate systems can be compressed into a multi-task ensemble. Teacher-student learning can be used to train the multi-task ensemble to emulate the diverse behaviours of the separate systems. A method has also been proposed to allow frame-level teacher-student learning to be used with lattice-free systems, which may be more diverse than lattice-based systems, but from which state cluster posteriors cannot be readily obtained. This is achieved by normalising the acoustic scores and using them with a KL-divergence criterion.

Chapter 6

Propagating different forms of information

The aim of teacher-student learning is for the student to emulate the behaviours of the teachers. The methods discussed in Section 4.3.3 and Chapter 5 aim to achieve this by propagating frame-level information of the state posteriors or acoustic scores, obtained at the outputs of NN acoustic models. It may also be possible to propagate other forms of information from the teachers to the student, to enable the student to better emulate the teachers' behaviours.

This chapter discusses two additional forms of information that can be propagated from the teachers to the student. Section 6.1 considers propagating over information about the hidden layer representations of the teachers. However, the hidden layer representations and state cluster posteriors are only intermediate representations, used to compute the hypothesis posteriors, and may not effectively capture information about the sequence-level behaviours of the teachers. It may be better for the student to directly learn from information about these sequence-level behaviours. Furthermore, with standard training methods, sequence-level training is often found to outperform frame-level training [86]. Section 6.2 investigates possible methods performing teacher-student learning at the sequence level.

6.1 Hidden layer information

Standard frame-level teacher-student learning using (4.15) propagates per-frame state cluster posterior information from the teachers to the student. As is described in Section 4.3.3, this may contain information about how difficult the teachers believe that each frame is to classify. In addition to this, there may be other forms of information that are useful to the student. When using a multi-layer NN topology for the acoustic model, each layer

performs a nonlinear projection of the previous layer's hidden representation. It may be beneficial for the student, if additional information is propagated about the hidden layer representation behaviours of the teachers. With this information, it may be useful for the student to develop hidden layer representations that are similar to those of the teachers. This can be interpreted as a form of regularisation, as the student is constrained to develop similar hidden representations as the teachers. The propagation of information about the hidden layer representations has been investigated for the purpose of model compression in [122] and for domain adaptation in [85].

When training a student to emulate only a single teacher, work in [122] examines propagating hidden layer representation information, by minimising the Mean Squared Error (MSE) between the teacher's hidden representations and a linear transformation of the student's hidden representations,

$$\mathcal{F}^{\text{MSE}}_{\text{hid-TS}}\left(\boldsymbol{\Theta}_{1:k'}, \boldsymbol{\Xi}\right) = \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{I}\left[\varphi_i\left(\mathbf{o}_t|\boldsymbol{\Phi}_{1:k}\right) - \varphi_i\left(\mathbf{o}_t|\boldsymbol{\Theta}_{1:k'}, \boldsymbol{\Xi}\right)\right]^2. \tag{6.1}$$

Here, I is the dimension of the teacher's hidden layer, $\varphi_i\left(\mathbf{o}_t|\boldsymbol{\Phi}_{1:k}\right)$ represents the activation of the ith node in the kth hidden layer of the teacher, and $\varphi_i\left(\mathbf{o}_t|\boldsymbol{\Theta}_{1:k'}, \boldsymbol{\Xi}\right)$ are the activations after applying a linear transformation with parameters of $\boldsymbol{\Xi}$ to the activations of the k'th hidden layer of the student, where $\boldsymbol{\Phi}_{1:k}$ and $\boldsymbol{\Theta}_{1:k'}$ are the teacher's and student's model parameters from the inputs up to the kth and k'th layers respectively. If the teacher and student have the same number of hidden layers, then it may be useful to have $k = k'$. However, in general, the student is allowed to have a different number of hidden layers than the teacher. The linear transformation, with parameters of $\boldsymbol{\Xi}$, allows the student and teacher to use different hidden layer dimensions. This criterion encourages the student to learn a hidden representation that is linearly related to that of the teacher. The criterion is minimised with respect to both $\boldsymbol{\Theta}_{1:k'}$ and $\boldsymbol{\Xi}$. The student can be trained by interpolating together $\mathcal{F}^{\text{MSE}}_{\text{hid-TS}}$ with the standard teacher-student learning criterion in (4.15), or by using $\mathcal{F}^{\text{MSE}}_{\text{hid-TS}}$ to perform layer-wise pre-training of the student [122]. Information can also be propagated between multiple hidden layers of the teacher and student.

When using an ensemble of teachers that are trained independently from each other, the individual teachers may develop different hidden representations. One possible way to extend $\mathcal{F}^{\text{MSE}}_{\text{hid-TS}}$ to allow for multiple teachers is to have a separate linear transformation of the student's hidden activations for each teacher,

$$\mathcal{F}^{\text{MSE}}_{\text{hid-TS}}\left(\boldsymbol{\Theta}_{1:k'}, \boldsymbol{\Xi}^1, \cdots, \boldsymbol{\Xi}^M\right) = \frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\lambda_m\sum_{i=1}^{I^m}\left[\varphi_i\left(\mathbf{o}_t|\boldsymbol{\Phi}^m_{1:k}\right) - \varphi_i\left(\mathbf{o}_t|\boldsymbol{\Theta}_{1:k'}, \boldsymbol{\Xi}^m\right)\right]^2, \tag{6.2}$$

where the interpolation weights satisfy $\lambda_m \geq 0$ and can be used to tune the contribution of each teacher when training the student. Here, $\boldsymbol{\Xi}^m$ are the parameters of the linear transformation, taking the student's hidden layer activations as input, and producing an output with the dimension of the hidden layer of the mth teacher. This criterion is minimised with respect to the student's model parameters up to the k'th hidden layer, $\boldsymbol{\Theta}_{1:k'}$, and also the multiple linear transformations, $\boldsymbol{\Xi}^1, \cdots, \boldsymbol{\Xi}^M$. Training multiple linear transformations may lead to a large number of parameters, which can be difficult to train to generalise well, with limited training data.

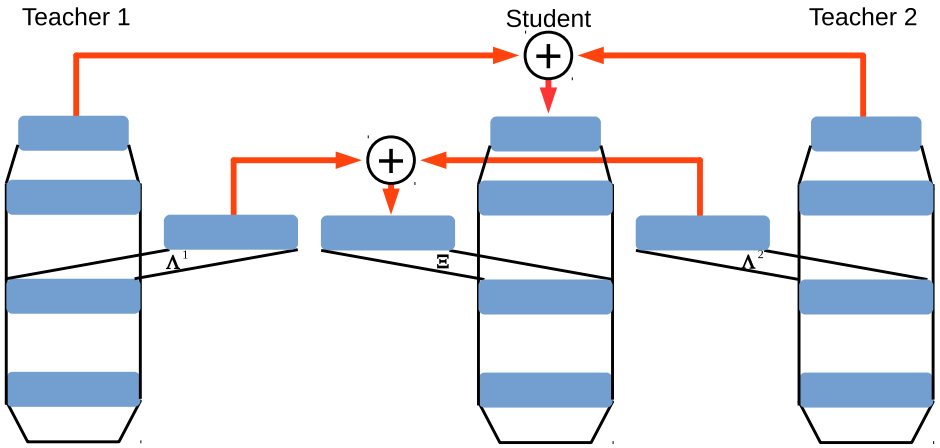


Fig. 6.1 Propagating hidden layer posterior information. Softmax output layers with parameters of $\boldsymbol{\Lambda}^1$ and $\boldsymbol{\Lambda}^2$ are trained to obtain posteriors from the hidden layers of the teachers. These hidden layer posteriors are then used to train the student, together with an additional softmax output layer with parameters of $\boldsymbol{\Xi}$.

Rather than training the student to develop hidden representations that are linearly related to those of the teachers, this thesis instead proposes to propagate hidden representation information within a probabilistic framework. Going from the input to the output of the teachers' NNs, each successive layer may project the hidden representation into a space where classification can be more easily performed. The student can be trained to develop hidden representations that express a similar difficulty of classifying each frame as the teachers' hidden representations.

This thesis proposes to achieve this as follows, and is illustrated in Figure 6.1. Single-layer softmax output layers with parameters of $\boldsymbol{\Lambda}^m$ are placed at the teachers' kth hidden layers. Using these new hidden layer outputs as state cluster posteriors, $P\left(s|\mathbf{o}_t, \boldsymbol{\Phi}^m_{1:k}, \boldsymbol{\Lambda}^m\right)$, the $\boldsymbol{\Lambda}^m$ parameters are trained by maximising the cross-entropy criterion of

$$\mathcal{F}_{\text{CE}}\left(\boldsymbol{\Lambda}^m\right) = \sum_{t=1}^{T} \log P\left(s_t^{\text{ref}} \middle| \mathbf{o}_t, \boldsymbol{\Phi}^m_{1:k}, \boldsymbol{\Lambda}^m\right). \tag{6.3}$$

The existing lower layer parameters of the teachers, $\mathbf{\Phi}^m_{1:k}$, are not updated here. The $\mathbf{\Lambda}^m$ parameters can also be trained toward a sequence discriminative criterion. A single-layer softmax output layer with parameters of Ξ is placed at the k'th hidden layer of the student. Both the student's lower layer parameters, $\mathbf{\Theta}_{1:k'}$, and Ξ are then trained by minimising the KL-divergence between the state cluster posteriors of these hidden layer outputs of the teachers and the student,

$$\mathcal{F}^{\text{KL}}_{\text{hid-TS}}\left(\mathbf{\Theta}_{1:k'}, \Xi\right) = -\sum_{t=1}^{T}\sum_{s\in\mathcal{T}}\sum_{m=1}^{M} \lambda_m P\left(s|\mathbf{o}_t, \mathbf{\Phi}^m_{1:k}, \mathbf{\Lambda}^m\right) \log P\left(s|\mathbf{o}_t, \mathbf{\Theta}_{1:k'}, \Xi\right), \quad (6.4)$$

where the interpolation weights satisfy $0 \leq \lambda_m \leq 1$ and $\sum_m \lambda = 1$, and can be used to tune the contribution of each teacher during training. The combined hidden layer posteriors of the teacher ensemble, $\sum_m \lambda_m P\left(s|\mathbf{o}_t, \mathbf{\Phi}^m_{1:k}, \mathbf{\Lambda}^m\right)$, are the targets used to train the student. These targets may propagate over hidden layer posterior information about how difficult frames are to classify, based on the hidden representations of the teachers.

Unlike in $\mathcal{F}^{\text{MSE}}_{\text{hid-TS}}$, a single additional student output layer with parameters of Ξ can be used for all of the teachers together in $\mathcal{F}^{\text{KL}}_{\text{hid-TS}}$. Therefore, the number of parameters that need to be updated when training the student using $\mathcal{F}^{\text{KL}}_{\text{hid-TS}}$ does not vary with the number of teachers. Furthermore, the KL-divergence criterion of $\mathcal{F}^{\text{KL}}_{\text{hid-TS}}$ is similar to the standard form of frame-level teacher-student learning in (4.15). Therefore, it may be simpler to integrate the propagation of hidden layer information using $\mathcal{F}^{\text{KL}}_{\text{hid-TS}}$ into an existing teacher-student learning infrastructure. This section has described propagating over hidden layer information from only one of the hidden layers of each of the teachers. It is possible to extend these methods to propagate information from multiple hidden layers of each teacher.

6.2 Sequence-level information

Section 4.3.3 and Chapter 5 discuss teacher-student learning using frame-level criteria. These propagate frame-level posterior information from the teachers to the student, and do not take into account the sequential nature of the data. Therefore, not all information about the sequence-level behaviours of the teachers may be effectively propagated to the student. Furthermore, standard frame-level teacher-student learning using (4.15) constrains the forms of diversities that are allowed within the ensemble, by requiring that all systems have the same set of state clusters. This in turn requires that all systems use the same HMM topology, context-dependence, and set of sub-word units. The proposed frame-level criterion of (5.5) allows for a diversity of state cluster sets. However, the frame-level posteriors do not take into account the alignment and language models, and therefore any diversity in these models

is not propagated over to the student. Frame-level teacher-student learning also requires all systems to produce state cluster posteriors, and needs to be modified to be used with lattice-free systems, as is discussed in Section 5.2.

This section discusses a generalisation of the teacher-student learning framework, where information is propagated at the sequence level. Previous work with standard training methods has shown that sequence-level training can often outperform frame-level training [86]. This section proposes to train the student at the sequence-level, to emulate the sequence-level behaviours of the teachers. Sequence-level teacher-student learning is originally proposed by the author of this thesis in [154]. Section 6.2.1 outlines the general aim of teacher-student learning, for the student to emulate the teachers' behaviours and produce similar recognition hypotheses. One possible criterion to achieve this is to minimise the KL-divergence between word sequence posteriors. This criterion is general, in that it does not place any constraints on the forms of diversities that are allowed in the ensemble, other than the need for the systems to produce hypothesis posteriors. However, the derivative of this criterion can be expensive to compute. In Section 6.2.2, an alternative criterion of a KL-divergence between lattice arc sequence posteriors is proposed, with a derivative that can be computed more efficiently. Section 6.2.3 discusses the particular case of marking the arcs with state clusters. This further improves the simplicity and efficiency of computing the derivative. However, this requires that all systems use the same set of state clusters, limiting the diversity that the ensemble is allowed to have. Finally, Section 6.2.4 investigates marking the arcs with logical context-dependent states. This allows for a diversity of state cluster sets, while preserving the same simplicity and efficiency of the criterion derivative computation as marking the arcs with state clusters.

6.2.1 Sequence-level teacher-student learning

Teacher-student learning aims for the student to emulate the combined teacher ensemble behaviour, by minimising a distance between the teachers and the student, expressed in (4.14). In ASR, what is ultimately intended is for the student to produce a similar recognition hypothesis as the combined ensemble. The frame-level teacher-student learning methods discussed in Section 4.3.3 and Chapter 5 aim to achieve this by minimising the distance between per-frame state posteriors. However, the frame-level information propagated in these approaches may not effectively capture the behaviours of the teachers at the sequence level. Instead, it may be better for the student to directly learn from the teachers' sequence-level behaviours.

Teacher-student learning in ASR aims for the student to produce a recognition hypothesis that is similar to that produced by the combined teacher ensemble. In the general MBR

decoding framework, the 1-best hypothesis of the combined ensemble can be obtained using MBR combination decoding of (3.29), repeated here as

$$\boldsymbol{\omega}^{\widehat{\boldsymbol{\Phi}}*} = \arg\min_{\boldsymbol{\omega}'} \sum_{\boldsymbol{\omega}} \mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right) P\left(\boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right). \tag{6.5}$$

One possible method of obtaining the combined hypothesis posteriors is to take a weighted average,

$$P\left(\boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) = \sum_{m=1}^{M} \lambda_m P\left(\boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right), \tag{6.6}$$

where the interpolation weights satisfy $0 \leq \lambda_m \leq 1$ and $\sum_m \lambda_m = 1$.

The most straight forward method of training the student may be to minimise a distance measure, $\mathbb{D}$, between the 1-best hypotheses of the teachers and student,

$$\mathcal{F}\left(\boldsymbol{\Theta}\right) = \mathbb{D}\left\{\boldsymbol{\omega}^{\widehat{\boldsymbol{\Phi}}*} \middle\| \boldsymbol{\omega}^{\boldsymbol{\Theta}*}\right\}. \tag{6.7}$$

Here, a sum over utterances in the training data is omitted for brevity. This is similar to cross-adaptation [43], discussed in Section 4.3.1, and lightly-supervised training [89]. In these methods, the 1-best hypotheses of one system are used as the transcriptions to train another system.

As is discussed in Section 4.3.3, one of the benefits that frame-level teacher-student learning may have is that the information propagated from the teachers may convey how difficult the teachers believe that each frame is to classify. Analogously at the sequence-level, it may be beneficial for the student, if information about the difficulty of classifying each utterance is propagated. Using only the 1-best hypotheses does not convey any information about classification difficulty.

Such information can be propagated by considering the competing hypotheses. From the decoding criterion of (6.5), it can be seen that the expected risk for each hypothesis, $\boldsymbol{\omega}'$, is $\sum_{\boldsymbol{\omega}} \mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right) P\left(\boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)$. One possible method to propagate information about the competing hypotheses is to minimise a distance, $\mathbb{D}$, between the expected risks over the range of competing hypotheses,

$$\mathcal{F}\left(\boldsymbol{\Theta}\right) = \sum_{\boldsymbol{\omega}'} \mathbb{D}\left\{\sum_{\boldsymbol{\omega}} \mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right) P\left(\boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) \middle\| \sum_{\boldsymbol{\omega}} \mathcal{L}\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right) P\left(\boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right)\right\}. \tag{6.8}$$

This form of criterion may propagate information about the expected risks of the competing hypotheses. It is interesting to consider the possible distance measures, $\mathbb{D}$, that may be used here. One possible option is to use the mean squared error. Another possible option, similar to

the trick proposed in Section 5.2, is to convert the expected risks to something that resembles a probability distribution. If the risk, $\mathcal{L}$ is non-negative, such as when using a minimum edit distance, then the expected risk can be made to resemble a probability distribution over hypotheses through normalisation. A KL-divergence can then be used as a distance measure. The investigation of such criteria is left for future work.

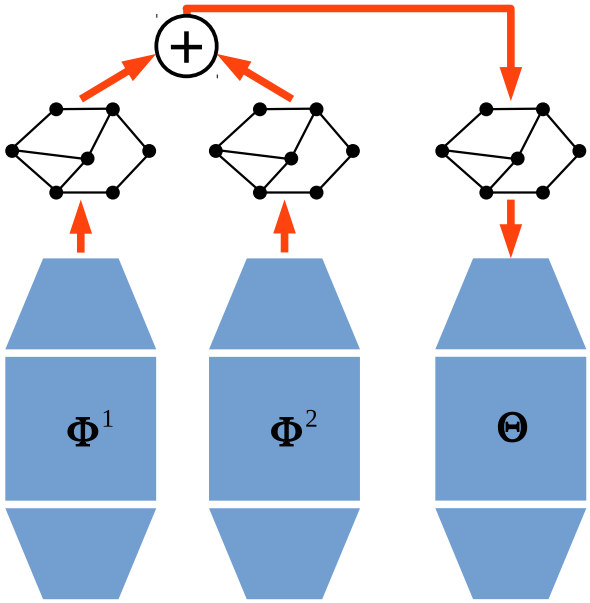


Fig. 6.2 Sequence-level teacher-student learning. Sequence-level information from the teachers are combined and propagated to the student.

The criterion of (6.8) can be simplified by considering the MAP risk function of (2.10), repeated here as

$$\mathcal{L}_{\text{MAP}}\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right) = 1 - \delta\left(\boldsymbol{\omega}, \boldsymbol{\omega}'\right). \tag{6.9}$$

This leads to minimising a distance between hypothesis posteriors,

$$\mathcal{F}\left(\boldsymbol{\Theta}\right) = \sum_{\boldsymbol{\omega}} \mathbb{D}\left\{P\left(\boldsymbol{\omega}\middle|\mathbf{O}_{1:T}, \hat{\boldsymbol{\Phi}}\right)\middle\|P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right)\right\}. \tag{6.10}$$

A commonly used measure of the distance between probability distributions is the KL-divergence, where

$$\mathbb{D}\left\{P\left(\boldsymbol{\omega}\middle|\mathbf{O}_{1:T}, \hat{\boldsymbol{\Phi}}\right)\middle\|P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right)\right\} = p\left(\mathbf{O}_{1:T}\right) P\left(\boldsymbol{\omega}\middle|\mathbf{O}_{1:T}, \hat{\boldsymbol{\Phi}}\right) \log \frac{P\left(\boldsymbol{\omega}\middle|\mathbf{O}_{1:T}, \hat{\boldsymbol{\Phi}}\right)}{P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right)}. \tag{6.11}$$

Using this, this thesis proposes that the student can be trained by minimising the KL-divergence between hypothesis posteriors,

$$\mathcal{F}^{\text{word}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right) = -\sum_{\boldsymbol{\omega}} P\left(\boldsymbol{\omega}\middle|\mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) \log P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right). \tag{6.12}$$

This propagates word sequence posterior information from the teachers to the student, which may convey information about the sequence-level behaviours of the teachers, and also about how difficult the teachers believe that each utterance is to classify.

The $\mathcal{F}_{\text{MMI}}$ criterion in (2.81) can be expressed as the minimisation of a KL-divergence,

$$\mathcal{F}_{\text{MMI}}\left(\boldsymbol{\Theta}\right) = -\sum_{\boldsymbol{\omega}} \delta\left(\boldsymbol{\omega}, \boldsymbol{\omega}^{\text{ref}}\right) \log P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right). \tag{6.13}$$

Comparing (6.13) with (6.12), it can be seen that the difference between these criteria is in the form of targets. Because of the similarity between these two criteria, it is simple to interpolate them together when training the student,

$$\mathcal{F} = \chi\mathcal{F}_{\text{MMI}}\left(\boldsymbol{\Theta}\right) + \left(1-\chi\right)\mathcal{F}^{\text{word}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right), \tag{6.14}$$

where the interpolation weight satisfies $0 \leq \chi \leq 1$. This is analogous to the interpolation between the cross-entropy and frame-level teacher-student learning criteria in (4.20) when training at the frame level. Interpolating together the $\mathcal{F}_{\text{MMI}}$ and $\mathcal{F}^{\text{word}}_{\text{seq-TS}}$ criteria allows information about the manual transcriptions to be used when training the student.

The criterion of $\mathcal{F}^{\text{word}}_{\text{seq-TS}}$ is general, in the sense that it does not place any constraints on the topologies and design choices of the systems used for the teachers and the student, other than that all systems must produce hypothesis posteriors. Therefore, a wide range of system architectures can be used for the ensemble and student, as long as the criterion derivative with respect to the student model parameters can be computed for gradient-based training. However, this criterion cannot be used directly with ASR systems that operate as discriminative functions, where hypothesis posteriors are not explicitly produced, such as with support vector machines.

The hybrid NN-HMM system, considered in the earlier chapters, has been shown to produce reasonable ASR performance [56]. This section therefore considers how a hybrid NN-HMM student can be trained. The derivative of $\mathcal{F}^{\text{word}}_{\text{seq-TS}}$ with respect to this student

acoustic model's pre-softmax activations, $z_{st}^{(K+1)}$, is

$$\frac{\partial \mathcal{F}_{\text{seq-TS}}^{\text{word}}(\boldsymbol{\Theta})}{\partial z_{st}^{(K+1)}} = \kappa \left[P\left(s_t^{\boldsymbol{\Theta}} = s \middle| \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) - \sum_{\boldsymbol{\omega} \in \mathbb{A}} P\left(s_t^{\boldsymbol{\Theta}} = s \middle| \boldsymbol{\omega}, \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) P\left(\boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) \right], \tag{6.15}$$

where κ is the acoustic scaling factor and $\mathbb{A}$ is the set of competing hypotheses, which may be represented by a lattice. Here, the superscript in $s_t^{\boldsymbol{\Theta}}$ emphasises that the derivative is computed over the student's set of state clusters, as this criterion allows for a diversity of state cluster sets. The first term in the derivative, $P\left(s_t^{\boldsymbol{\Theta}} \middle| \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right)$, is the same denominator term that is found in the $\mathcal{F}_{\text{MMI}}$ derivative of (2.111). This can be computed using a forward-backward operation over a lattice of competing hypotheses, generated by the student, as is described in Section 2.5.1. The second term, $\sum_{\boldsymbol{\omega}} P\left(s_t^{\boldsymbol{\Theta}} \middle| \boldsymbol{\omega}, \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) P\left(\boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)$, requires the computation of $P\left(\boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)$ and $P\left(s_t^{\boldsymbol{\Theta}} \middle| \boldsymbol{\omega}, \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right)$. Each of these can be computed using different forms of forward-backward operations over the lattices of the teachers and student. However, these need to be computed and stored for every competing hypothesis, $\boldsymbol{\omega}$. It can therefore be computationally expensive to train the student in this manner when there are many competing hypotheses. This computational cost can be limited by only considering a finite number of competing hypotheses in the sum over hypotheses in (6.15). These can be limited to an n-best list, or a Monte Carlo approximation to the derivative can be taken with a finite number of hypothesis samples, similarly to [132].

It may be an interesting topic for future research, to investigate whether it is possible to use a forward-backward algorithm to compute $\sum_{\boldsymbol{\omega}} P\left(\boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) P\left(s_t^{\boldsymbol{\Theta}} \middle| \boldsymbol{\omega}, \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right)$ directly, without needing to sum over the hypotheses. The difficultly in developing such a forward-backward algorithm is two-fold. First, the $P\left(\boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)$ and $P\left(s_t^{\boldsymbol{\Theta}} \middle| \boldsymbol{\omega}, \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right)$ terms are computed using the acoustic, alignment, and language scores from different systems. Second, a hypothesis, $\boldsymbol{\omega}$, does not specify the start and end times of each word. Therefore, all possible time alignments of the same word sequence hypothesis in the teachers' and student's lattices need to be considered.

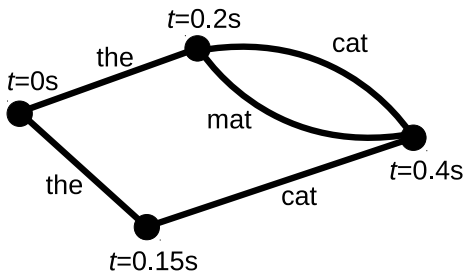
6.2.2 Arc sequence posteriors

Rather than limiting the hypotheses that are considered in the derivative computation, another solution is to modify the training criterion, to eliminate the need to sum over sequences when computing the derivative. This thesis proposes that one possible criterion is to minimise the KL-divergence between posteriors of lattice arc sequences, instead of word sequences,

$$\mathcal{F}_{\text{seq-TS}}^{\text{arc}}(\boldsymbol{\Theta}) = -\sum_{\boldsymbol{\omega}} \sum_{\mathbf{a}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) \log P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right), \tag{6.16}$$

the cat
the mat

(a) Word sequences do not specify the start and end times of words.



(b) Word arc sequences with word start and end times can be obtained from a word-marked lattice.

Fig. 6.3 Difference between word sequences and lattice arc sequences.

where $\mathcal{G}_{\boldsymbol{\omega}}$ here represents the set of lattice arc sequences, $\mathbf{a}_{1:T}$, that can represent the word sequence $\boldsymbol{\omega}$. The difference between arcs, $\mathbf{a}_{1:T}$, and words, $\boldsymbol{\omega}_{1:L}$, is that unlike words, arcs have defined start and end times. This difference is illustrated in Figure 6.3. The arc sequence posteriors are related to the word sequence posteriors as

$$P\left(\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\boldsymbol{\Theta}\right) = \sum_{\mathbf{a}_{1:T}\in\mathcal{G}_{\boldsymbol{\omega}_{1:L}}} P\left(\mathbf{a}_{1:T},\boldsymbol{\omega}_{1:L}|\mathbf{O}_{1:T},\boldsymbol{\Theta}\right). \tag{6.17}$$

The word sequence posterior is the sum over the posteriors of all arc sequences that consider all possible time alignments and pronunciations of the words. The arc sequence KL-divergence is in fact an upper bound to the word sequence KL-divergence. A proof of this can be found in Appendix A. Therefore, minimising $\mathcal{F}^{\text{arc}}_{\text{seq-TS}}$ minimises an upper bound to $\mathcal{F}^{\text{word}}_{\text{seq-TS}}$.

As is described in Section 2.5.1, the lattice arcs can be marked with words, sub-word units, or states, leading to different possible criteria. The joint posterior between arc and word sequences is considered in $\mathcal{F}^{\text{arc}}_{\text{seq-TS}}$, since for certain choices of arc markings, such as states clusters or phones, the arc sequence may not uniquely determine the word sequence, because of the possibility of homophonic words. With homophonic words, the same arc sequence with different word sequences may have different language model probabilities.

The derivative of $\mathcal{F}^{\text{arc}}_{\text{seq-TS}}$ is

$$\frac{\partial\mathcal{F}^{\text{arc}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right)}{\partial z^{(K+1)}_{st}} = \kappa\left[P\left(s^{\boldsymbol{\Theta}}_t = s\middle|\mathbf{O}_{1:T},\boldsymbol{\Theta}\right) - \sum_{a_t} P\left(s^{\boldsymbol{\Theta}}_t = s\middle|a_t,\mathbf{O}_{1:T},\boldsymbol{\Theta}\right) P\left(a_t\middle|\mathbf{O}_{1:T},\widehat{\boldsymbol{\Phi}}\right)\right]. \tag{6.18}$$

The first term, $P\left(s^{\boldsymbol{\Theta}}_t\middle|\mathbf{O}_{1:T},\boldsymbol{\Theta}\right)$, is again the denominator term, present in the $\mathcal{F}_{\text{MMI}}$ derivative of (2.111). Unlike the derivative of $\mathcal{F}^{\text{word}}_{\text{seq-TS}}$ in (6.15), there is no longer a sum over sequences in the second term of $\sum_{a_t} P\left(s^{\boldsymbol{\Theta}}_t\middle|a_t,\mathbf{O}_{1:T},\boldsymbol{\Theta}\right) P\left(a_t\middle|\mathbf{O}_{1:T},\widehat{\boldsymbol{\Phi}}\right)$, because the arcs have defined

start and end times. Instead, the sum here is over the arcs that overlap in time with the frame for which the derivative is computed. The number of arcs is often fewer than the number of word sequence hypotheses. As such, the cost of computing (6.18) may be much less than that for (6.15).

The second term in the derivative of (6.18) requires the computation of $P\left(a_t\middle|\mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)$ and $P\left(s_t^{\boldsymbol{\Theta}}\middle|a_t, \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right)$ for each arc, a_t. The $P\left(a_t\middle|\mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)$ term can be computed using a forward-backward operation over the teachers' lattices that have been marked with the chosen acoustic unit. To compute the $P\left(s_t^{\boldsymbol{\Theta}}\middle|a_t, \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right)$ term, each arc can be expanded into a state cluster-marked lattice, over which a forward-backward operation can be performed. The second term in the derivative of (6.18) therefore requires two levels of forward-backward operations to compute.

The lattices can either be pruned in a lattice-based framework, or unpruned in a lattice-free framework, as is discussed in Section 2.5. In the lattice-free framework, a sub-word unit language model is often used, instead of a word-level language model, to reduce the computational cost when performing training. As such, the lattices are often generated by only composing the alignment model, $\mathbb{H}$, and context-dependence, $\mathbb{C}$, graphs. In such a configuration, the lattice arcs cannot be marked with words when training the student. On the other hand, marking the arcs with sub-word units requires all systems to use the same set of sub-word units. Marking the arcs with states further requires all systems to use the same set of states, context-dependence, and HMM topology.

6.2.3 State cluster sequence posteriors

Marking the arcs with state clusters further simplifies the criterion derivative computation. This leads to a criterion of minimising a KL-divergence between state cluster sequence posteriors,

$$\mathcal{F}_{\text{seq-TS}}^{\text{state}}\left(\boldsymbol{\Theta}\right) = -\sum_{\boldsymbol{\omega}}\sum_{\mathbf{s}_{1:T}\in\mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{s}_{1:T}, \boldsymbol{\omega}\middle|\mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)\log P\left(\mathbf{s}_{1:T}, \boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right). \tag{6.19}$$

This method has been investigated by the author of this thesis in [154]. The state cluster sequence uniquely determines the state cluster at each time step. As such, with this choice of arc marking, $P\left(s_t^{\boldsymbol{\Theta}}\middle|a_t, \mathbf{O}, \boldsymbol{\Theta}\right)$ in (6.18) becomes a δ-function, and the criterion derivative reduces to

$$\frac{\partial\mathcal{F}_{\text{seq-TS}}^{\text{state}}\left(\boldsymbol{\Theta}\right)}{\partial z_{st}^{(K+1)}} = \kappa\left[P\left(s_t = s|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) - P\left(s_t = s\middle|\mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)\right]. \tag{6.20}$$

Here, the second term, $P\left(s_t \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)$, only requires a single level of forward-backward operations over the teachers' lattices to compute. As opposed to this, the equivalent term in the derivative of (6.18) requires two levels of forward-backward operations to compute, when using other choices of arc markings. This difference is analogous to that between the state cluster marked $\mathcal{F}_{\text{sMBR}}$ criterion [48, 115], and the more general minimum phone error and minimum word error criteria [117]. Teacher-student learning using arcs marked with state clusters is therefore simpler to implement and may be less computationally expensive to using during training than other choices of arc markings.

One possible method to obtain the target sequence posteriors is to perform a sum combination of the teachers' posteriors,

$$P\left(\mathbf{s}_{1:T}, \boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) = \sum_{m=1}^{M} \lambda_m P\left(\mathbf{s}_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right). \tag{6.21}$$

Using this method of target combination, the second term in the derivative of (6.20) is also a linear interpolation of the contributions from each teacher,

$$P\left(s_t \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) = \sum_{m=1}^{M} \lambda_m P\left(s_t | \mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right). \tag{6.22}$$

This requires a separate forward-backward operation over the lattices of each teacher in the ensemble.

It is also possible to obtain the targets as a product combination over the teachers' posteriors. Since the product is taken over the posteriors of state sequences, this can be shown to be equivalent to a frame-level combination, using a product over scaled observation likelihoods [78],

$$P\left(\mathbf{s}_{1:T}, \boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) = \frac{1}{Z\left(\mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)} \prod_{m=1}^{M} P^{\lambda_m}\left(\mathbf{s}_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right) \tag{6.23}$$

$$= \frac{1}{Z\left(\mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)} \prod_{m=1}^{M} \left[P^{\gamma}\left(\boldsymbol{\omega}\right) \prod_{t=1}^{T} P^{\gamma}\left(s_t | s_{t-1}\right) \mathcal{A}^{\kappa}\left(\mathbf{o}_t, s_t, \boldsymbol{\Phi}^m\right) \right]^{\lambda_m} \tag{6.24}$$

$$= \frac{1}{Z\left(\mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)} P^{\gamma}\left(\boldsymbol{\omega}\right) \prod_{t=1}^{T} P^{\gamma}\left(s_t | s_{t-1}\right) \prod_{m=1}^{M} \mathcal{A}^{\kappa\lambda_m}\left(\mathbf{o}_t, s_t, \boldsymbol{\Phi}^m\right), \tag{6.25}$$

if $\sum_{m=1}^{M} \lambda_m = 1$. Here, $Z\left(\mathbf{O}_{1:T}, \widehat{\mathbf{\Phi}}\right)$ ensures that the combined posteriors are normalised. Using this product combination, the second term in the derivative of (6.20), $P\left(s_t \middle| \mathbf{O}_{1:T}, \widehat{\mathbf{\Phi}}\right)$, can be computed with a forward-backward operation over just a single lattice for the whole ensemble. The acoustic scores for this combined lattice are obtained through a frame-level combination of the teachers' scaled observation likelihoods. Using a product combination of the target posteriors is therefore less computationally expensive during training than using a sequence-level sum combination of (6.21).

Although a KL-divergence between state cluster sequence posteriors simplifies the derivative computation, it also foregoes much of the freedom in the forms of diversities in the ensemble that is allowed by other choices of arc markings. In particular, all members of the ensemble are restricted to use the same set of state clusters as the student. This in turn requires identical sets of sub-word units, HMM topologies, and context-dependencies.

6.2.4 Logical context-dependent state sequence posteriors

One possible method to allow the systems to use different sets of state clusters is to mark the lattice arcs with words or sub-word units. However, these arc markings require two levels of forward-backward operations when computing the criteria derivatives. This section considers a method of allowing for a diversity of state cluster sets with sequence-level teacher-student learning, while preserving the simplicity and computational efficiency of only requiring a single level of forward-backward operations to compute the criterion derivative.

In Section 5.1.1, frame-level teacher-student learning is generalised to allow for a diversity of state cluster sets, by minimising a KL-divergence between per-frame logical context-dependent state posteriors. Analogously, this thesis proposes to allow for different sets of state clusters in sequence-level teacher-student learning, by marking the arcs with logical context-dependent states, instead of state clusters. This leads to a criterion of minimising the KL-divergence between logical Context-Dependent (CD) state sequence posteriors,

$$\mathcal{F}^{\text{CD}}_{\text{seq-TS}}\left(\mathbf{\Theta}\right) = -\sum_{\boldsymbol{\omega}} \sum_{\mathbf{c}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{c}_{1:T}, \boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\mathbf{\Phi}}\right) \log P\left(\mathbf{c}_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \mathbf{\Theta}\right). \tag{6.26}$$

For a hybrid NN-HMM system and using this arc marking, the sequence posteriors can be computed similarly to (2.47), as

$$P\left(\mathbf{c}_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \mathbf{\Theta}\right) = \frac{P^{\gamma}\left(\boldsymbol{\omega}\right) \prod_{t=1}^{T} P^{\gamma}\left(c_t | c_{t-1}\right) \mathcal{A}^{\kappa}\left(\mathbf{o}_t, c_t, \mathbf{\Theta}\right)}{\sum_{\boldsymbol{\omega}'} P^{\gamma}\left(\boldsymbol{\omega}'\right) \sum_{\mathbf{c}'_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}'}} \prod_{t=1}^{T} P^{\gamma}\left(c'_t | c'_{t-1}\right) \mathcal{A}^{\kappa}\left(\mathbf{o}_t, c'_t, \mathbf{\Theta}\right)}, \tag{6.27}$$

if $\mathbf{c}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}$, and zero otherwise. Here, the alignment model, $P\left(c_t|c_{t-1}\right)$, is over logical context-dependent states. The scaled observations likelihoods for all logical context-dependent states within the same state cluster are tied according to (2.29), repeated here as

$$\mathcal{A}\left(\mathbf{o}_t, c, \boldsymbol{\Theta}\right) = \mathcal{A}\left(\mathbf{o}_t, s^{\boldsymbol{\Theta}}, \boldsymbol{\Theta}\right) \quad \forall c : \mathcal{T}^{\boldsymbol{\Theta}}\left(c\right) = s^{\boldsymbol{\Theta}}. \tag{6.28}$$

The derivative of this criterion is

$$\frac{\partial \mathcal{F}^{\text{CD}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right)}{\partial z^{(K+1)}_{st}} = \kappa \left[P\left(s^{\boldsymbol{\Theta}}_t = s \middle| \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) - \sum_{c_t : \mathcal{T}^{\boldsymbol{\Theta}}(c_t) = s} P\left(c_t \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) \right]. \tag{6.29}$$

For computational efficiency, this can be re-expressed as a sum over intersect states, $\widehat{s}_t$,

$$\frac{\partial \mathcal{F}^{\text{CD}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right)}{\partial z^{(K+1)}_{st}} = \kappa \left[P\left(s^{\boldsymbol{\Theta}}_t = s \middle| \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) - \sum_{\widehat{s}_t \in \mathcal{G}^{\boldsymbol{\Theta}}_s} P\left(\widehat{s}_t \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right) \right], \tag{6.30}$$

formed by a Cartesian product of the decision trees of all the teachers and the student. Here, $\mathcal{G}^{\boldsymbol{\Theta}}_s$ represents the set of intersect states, $\widehat{s}_t$, that are contained within the student's state cluster, s. Similarly to the derivative of the $\mathcal{F}^{\text{state}}_{\text{seq-TS}}$ criterion in (6.20), the second term, $P\left(\widehat{s}_t \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)$, in the derivative here can also be computed using a single level of forward-backward operations over the teachers' lattices. Here, these lattice arcs are marked with intersect states. As opposed to this, marking the arcs with words or sub-word units requires two levels of forward-backward operations to compute the equivalent term in (6.18). Therefore, even though a diversity of state cluster sets can also be allowed by marking the arcs with words or sub-word units, it can be simpler to implement and possibly more computationally efficient to train the student when the lattice arcs are marked with logical context-dependent states, or intersect states.

If the same set of state clusters is used across all systems, it can be shown using a similar argument to that in Appendix A, that $\mathcal{F}^{\text{word}}_{\text{seq-TS}} \leq \mathcal{F}^{\text{state}}_{\text{seq-TS}} \leq \mathcal{F}^{\text{CD}}_{\text{seq-TS}}$. Equality, $\mathcal{F}^{\text{state}}_{\text{seq-TS}} = \mathcal{F}^{\text{CD}}_{\text{seq-TS}}$, is obtained if each state cluster sequence can be uniquely described by a single logical context-dependent state sequence. This can be achieved by, for example, having a different root node in the decision tree for each different centre phone.

As is discussed in Section 5.1.1, when performing frame-level teacher-student learning with a diversity of state cluster sets, the target posteriors need to be mapped between different sets of state clusters, using (5.9). The map is estimated with the approximation of (5.11). This approximation may lead to a loss of phonetic resolution in the targets. At the sequence level, marking the lattice arcs with logical context-dependent states, or intersect states, allows the targets of $P\left(\mathbf{c}_{1:T}, \boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \widehat{\boldsymbol{\Phi}}\right)$ in the criterion of (6.26) to be computed exactly, without

the need for any approximations. The surrounding context of a sub-word unit is known when given the state sequence. This avoids any degradation of the student that may result from any approximations.

6.3 Summary

This chapter has considered various forms of information that can be propagated from the teachers to the student. A method has been proposed to propagate information about the difficulty of classifying each frame, based on the hidden layer representations of the teachers. The teacher-student learning framework has also been generalised to the sequence level, allowing the propagation of sequence posterior information. One approached is to minimise the KL-divergence between word sequence posteriors of the teachers and student. However, the derivative of this criterion can be computationally expensive to compute. The student can instead be trained by minimising the KL-divergence between lattice arc sequence posteriors. The derivative of this criterion can be computed using two levels of forward-backward operation over the teachers' lattices. This form of criterion is shown to be an upper bound to a KL-divergence over word sequence posteriors. Marking the arcs with state clusters allows an even simpler and possibly more efficient derivative computation, requiring only a single level of forward-backward operations over the teachers' lattices. However, such an arc marking requires all systems to use the same set of state clusters, and therefore places restrictions on the allowed forms of ensemble diversities. Instead, the arcs can be marked with logical context-dependent states. This allows the ensemble to capture a diversity of state cluster sets, while still having a simple derivative computation with only a single level of forward-backward operations over the teachers' lattices.

Chapter 7

Experimental setup

This chapter describes the datasets and system configurations that were used in the experiments presented in this thesis.

7.1 Datasets

Table 7.1 Datasets.

Dataset	ID	Training set size (hours)
Babel Tok Pisin VLLP	207V	3
Augmented Multi-party Interaction	AMI-IHM	81
English broadcast news	HUB4	144
Multi-Genre Broadcast 3	MGB-3	275

The experiments in this thesis were conducted over a range of datasets. These provide a variety of test conditions, with different quantities of training data, recording environments, and speaking styles. The Augmented Multi-party Interaction (AMI) meeting transcription dataset [18] comprises audio data from a simulated workplace meeting interaction between multiple participants, in the English language. The Individual Headset Microphone (IHM) recordings were used, providing fairly clean speech. The *full corpus ASR partition* was used, containing approximately 81 hours of training data and a 9 hours *eval* set. The standard phonetic dictionary from Carnegie Mellon University[1] was used. Both trigram and 4-gram language models were trained on a combination of the AMI training set and Fisher English training part 1 (*LDC2004T19*) transcriptions. When performing recognition, decoding

[1] https://svn.code.sf.net/p/cmusphinx/code/trunk/cmudict

lattices were first generated using the trigram language model, then rescored using the 4-gram language model. The AMI-IHM dataset is used as the main task for experimentation in this thesis where possible, with the other tasks providing secondary supporting experimental evidence where needed.

The Tok Pisin language dataset (*IARPA-babel207b-v1.0e*) contains spontaneous conversational speech between two parties across a telephone channel, from the IARPA Babel programme [62]. The Very Limited Language Pack (VLLP) was used, containing approximately 3 hours of training data. This shall be referred to as 207V. The standard 10 hours development set was used for evaluation. The small training set size of 207V makes it ideal to quickly run preliminary concept tests. The aim of the Babel programme is to develop systems when constrained with limited resources, both in terms of the quantity of training data and the linguistic expertise. Toward this aim, the standard evaluation of the Option 2 data distribution, in which 207V belongs, is performed using a graphemic dictionary. This does not require linguistic expertise to obtain, as is discussed in Section 2.2.2. As such, a graphemic dictionary was used for this dataset. Work in [45] has shown that a graphemic Tok Pisin system performs competitively against a phonetic system. The use of a graphemic dictionary here provides a contrast against the phonetic dictionaries of the other datasets, to demonstrate the consistency of the trends in the experimental results. A trigram language model was trained on the training data transcriptions.

The HUB4 English broadcast news task comprises audio recordings of television and radio news programmes. The training set contains approximately 144 hours of audio data, from both the 1996 [53] and 1997 [39] releases. The 2.6 hours *eval03* test set was used for evaluation. The phonetic dictionary used was obtained from the LIMSI 1993 Wall Street Journal dictionary [46]. As is described in [158], this dictionary was extended with additional pronunciations from a text-to-speech system, and modified with manual corrections. A trigram language model was imported over from the *RT-04* system [84]. This had been trained on a combination of closed captions and manual transcriptions of the broadcasts, details of which are provided in [44].

The 2017 English Multi-Genre Broadcast 3 (MGB-3) [5] dataset comprises audio recordings from television programmes that have been broadcast by the BBC. This is a follow-up challenge to the original 2015 MGB challenge [6]. These audio recordings are from programmes of a variety of different genres, covering advice, children's, comedy, competition, documentary, events, and news shows. The transcriptions that are initially provided together with the dataset release are the closed captions that were broadcast together with the programmes. These tend to be fairly inaccurate, with both misaligned timing information and word errors. This is because the closed captions are often produced by people reiterating

what they hear from the broadcast shows into an ASR system. This may cause time delays, and what is reiterated may be a reformulation of what is actually said in the show, leading to word errors. The lightly supervised decoding and data selection methods described in [90] were used to select data for training. Here, an initial ASR system was first used to transcribe the training data. These ASR transcriptions were compared to the closed caption data, and confidence scores were computed. These were then used to re-align the closed captions and select a subset of the training data, for which the confidence that the closed captions are correct was above a threshold. These processes resulted in a final training set with approximately 275 hours of audio data, out of the full 375 hours that was initially provided. The 5.5 hours *dev17b* test set was used for evaluation. This test set was divided up into utterances, using a DNN-based segmenter, described in [149]. A phonetic dictionary was used, derived from Combilex [40]. A trigram language model was trained on the provided closed caption data.

7.2 Configurations

The experiments in this thesis were run using the Kaldi toolkit [114]. The hybrid NN-HMM system architecture was used for all datasets. Initial GMM-HMM systems were built. For 207V, the GMM-HMM was trained on a concatenation of multi-lingual bottleneck, PLP, pitch, and probability-of-voicing features [148], forming what shall be referred to as tandem features. The 207V dataset comprises a very limited quantity of training data, which is also noisy. In this setting, the multi-lingual bottleneck features, which were trained from data across the multiple Babel languages, were found to greatly benefit the ASR systems, over using just hand-crafted features. The AMI-IHM, HUB4, and MGB-3 dataset contain much more data than 207V, and have very different recording conditions. Following the standard Kaldi *s5* recipe, the GMM-HMM in AMI-IHM was trained on MFCC features. The GMM-HMM in MGB-3 was trained on PLP features. For HUB4, an existing uni-lingual bottleneck feature extractor was available, and was used to generate tandem features. These uni-lingual tandem features were found to be beneficial for the GMM-HMM. The GMM-HMMs for all datasets were trained using speaker adaptive training.

Forced alignments from these GMM-HMMs were used as targets for cross-entropy training of NN-HMMs, with the $\mathcal{F}_{\text{CE}}$ criterion of (2.80). Three types of NN acoustic model topologies were used in this thesis. The DNN and BLSTM models are described here, and a TDNN-LSTM model is later described in Chapter 11. Feed-forward DNN models with 4 layers of 1000 nodes for 207V, 6 layers of 2048 nodes for AMI-IHM, and 6 layers of 2000 nodes for HUB4 were used. All DNN models used sigmoid activation functions. NN acoustic

Table 7.2 Neural network topologies.

Dataset	Topology	Layers	Nodes per layer	Decision tree size
207V	DNN	4	1000	1000
AMI-IHM	DNN	6	2048	4000
	BLSTM	2	per direction: 1000 cells, 500 projections	
HUB4	DNN	6	2000	6000

models with longer temporal contexts were also trained. BLSTM models were used with the AMI-IHM dataset. These had 2 layers of 1000 cells, with the output linearly projected to a smaller 500 dimensions, in each temporal direction. The NN topologies are summarised is Table 7.2. In AMI-IHM and HUB4, these NN acoustic models used 40-dimensional Mel-scaled filterbank features as inputs. In 207V, the multi-lingual tandem features were used as inputs to the NN acoustic models, as these were found to give performance gains on this dataset. Temporal derivatives were appended to the features and a temporal context was spliced together, when used with a feed-forward DNN acoustic model. This led to a total temporal context of 9, 19, and 13 frames of input features for 207V, AMI-IHM, and HUB4 respectively.

Recognition was performed using MBR decoding of (2.9). Unless explicitly mentioned, ensemble combinations were performed using hypothesis-level MBR combination decoding of (3.29). Equal interpolation weights were used to combine the systems. The statistical significance of the results was measured using the matched pairs sentence segment word error test [49]. Different, but fixed language scaling factors were used for each dataset and for systems trained with different training criteria, as it was found that each criterion yielded acoustic scores with different dynamic ranges.

All datasets used triphone or trigrapheme contexts. The 3-state HMM topology shown in Figure 2.2a was used. The decision trees had 1000, 4000, 6000, and 9200 leaves for 207V, AMI-IHM, HUB4, and MGB-3 respectively. These trees were configured to have separate roots for each different centre phone. In the default Kaldi tree building process, a bottom-up re-clustering is applied to merge together leaves that result in small changes to the likelihood. This re-clustering was not performed in the experiments in this thesis, as it is convenient to compare decision trees with the same number of leaves, especially in the random forest experiments. The DNN model parameters were initialised using stacked-RBM pre-training [68] for AMI-IHM, following the standard Kaldi recipe. The model parameters were initialised with supervised pre-training toward the cross-entropy criterion for 207V and HUB4. These systems were then trained using the cross-entropy criterion, with a NewBob learning rate schedule [119]. Further sequence discriminative training was then performed

using a lattice-based implementation of the state-level MBR, $\mathcal{F}_{\text{sMBR}}$, criterion of (2.83), with a fixed learning rate.

Chapter 8

Experiments on ensemble generation and combination

This chapter investigates various approaches of generating and combining ensembles. In Chapter 3, a variety of methods are presented to generate an ensemble of systems, with different forms of diversities. Sections 8.1 to 8.5 presents experiments to investigate how each of these forms of diversities can contribute to the combined ensemble performance. A range of ensemble combination approaches, discussed in Section 3.4, are assessed in Section 8.6. However, these combination methods can be computationally expensive. Section 8.7 investigates reducing this computational cost by merging the hidden layers into a multi-task ensemble, as is described in Section 4.2.

8.1 Acoustic model diversity

A simple approach to generating an ensemble is to only use a diversity of acoustic models. This constrains the set of state clusters, set of sub-word units, and feature representations to be the same for all members of the ensemble. Such an ensemble is explored in this section. By constraining the acoustic model topology to be the same, a diversity of model parameter sets is explored. However, using only different sets of model parameters may limit the diversity that can be captured within the ensemble, as is described in Section 3.3.2. A diversity of acoustic model topologies is assessed, by combining an ensemble of DNN and BLSTM models.

8.1.1 Random initialisation

This section assesses the diversity that can be obtained when using different sets of model parameters, with the model topology fixed. As is described in Section 3.2.3, a simple method to generate an ensemble with different sets of model parameters is to perform multiple training runs, each starting from a different random parameter initialisation. In the AMI-IHM dataset, an ensemble of 4 DNN models was constructed, each beginning from a different random parameter initialisation. Each model was first pre-trained using the stacked-RBM method, then fine-tuned toward the cross-entropy criterion in (2.80). Finally, further sequence discriminative training using the $\mathcal{F}_{\text{sMBR}}$ criterion of (2.83) was performed.

Table 8.1 Cross-entropy and sequence trained random initialisation ensembles, in AMI-IHM. Each ensemble had 4 DNNs and was combined using hypothesis-level MBR combination decoding. Diversity and combination gains can be obtained using different random parameter initialisations.

	Single system WER (%)				Combined	
Training	mean	best	worst	std dev	WER (%)	Cross-WER (%)
$\mathcal{F}_{\text{CE}}$	28.4	28.3	28.5	0.08	27.6	11.4
+ $\mathcal{F}_{\text{sMBR}}$	25.7	25.6	25.8	0.10	24.9	11.8

Table 8.1 shows the ensemble performances. Here, hypothesis-level MBR combination decoding was used to combine the systems in the ensembles. The results show that beginning training from different random parameter initialisations allows the systems to develop diverse behaviours and combination gains. This diversity is obtained, even though the acoustic models had been pre-trained. The original motivation for performing pre-training was to reduce the sensitivity of the model to the initialisation [68]. However, the results show that even with pre-training, the initialisation can still significantly affect the model behaviour. The absolute differences between the WER performances of the systems within the ensemble are small, as is indicated by the WER standard deviation. Despite this, significant combination gains can still be obtained. The single system WER standard deviation therefore does not seem to be a reliable indicator of ensemble diversity. Instead, the diversity can be measured using the cross-WER of (3.64). This estimates the fraction of words that are recognised differently between the systems. The cross-WERs are measured to be significant fractions of the single system WERs, and therefore show significant differences between the 1-best

transcriptions of the systems. As a reference, the maximum possible cross-WER between two systems is approximately[1] the sum of their WERs.

Sequence training improves upon the single system performances over frame-level training, leading to a better combined performance. The cross-WER may suggest that sequence training slightly increases the ensemble diversity. In the lattice-based sequence training performed here, the lattices used for the computation of the criterion derivative were separately generated from each of the initial cross-entropy systems. These may have variations in their time alignments and hypotheses. This may allow the sequence-level systems to diverge from any bias that they may have toward the cross-entropy forced alignments.

8.1.2 Monte Carlo Dropout

Generating an ensemble from multiple random parameter initialisations requires multiple training runs. This can become computationally expensive during training as the ensemble size grows. The Monte Carlo Dropout method, discussed in Section 3.1.3, is one possible approach to obtaining multiple sets of model parameters, while only requiring a single training run. Monte Carlo Dropout ensembles were generated, each with a different Dropout rate, in AMI-IHM. The systems were again trained up to the sequence level, and Dropout was used in both the frame and sequence-level training stages. Each system was decoded 4 times with different random seeds used to sample the Dropout masks, to construct each ensemble. During recognition, the same Dropout rates were used as during training. These ensembles were combined using hypothesis-level MBR combination decoding of (3.29). It is also possible to not use Dropout when performing recognition [137]. As is described in Section 4.3.4, this can be interpreted as performing parameter-level combination of an ensemble. Table 8.2 compares these ensembles.

The results suggest that when the Dropout rate is small, the ensemble only has a small amount of diversity. This may be due to the form of the model parameter posterior of (3.16) that is imposed on the ensemble by Dropout. The model parameter posterior of (3.16) only has non-zero probability for a limited set of parameter values. This small diversity leads to limited combination gains, less than those obtained in the ensemble that was generated from multiple random parameter initialisations, in Table 8.1. Increasing the Dropout rate can lead to more diverse systems. However, at the same time, the systems are also more heavily regularised, as is discussed in Section 2.4.4, and the single ensemble member performance

[1]The maximum possible minimum edit distance between two systems is the sum of each of their minimum edit distances to the reference transcription. However, the WER and cross-WER are normalised by the lengths of the transcriptions that are used as references, which may vary. Therefore the sum of the WERs is not actually the upper limit of the cross-WER.

Table 8.2 Monte Carlo Dropout, in AMI-IHM. Each ensemble was generated by decoding a single DNN 4 times, with different Dropout mask samples, and was combined using MBR combination decoding. All systems were trained with the $\mathcal{F}_{\text{sMBR}}$ criterion. The ensembles exhibit only small diversity.

	WER (%) w Dropout				WER (%)	Combined	cross-
Dropout rate	mean	best	worst	std dev	w/o Dropout	WER (%)	WER (%)
0.0	25.7	-	-	-	25.7	-	-
0.1	26.1	26.1	26.1	0.00	25.5	25.8	8.3
0.2	28.1	28.0	28.2	0.08	26.9	27.5	11.0
0.3	31.8	31.7	31.8	0.05	28.9	30.7	15.2

degrades. This demonstrates how the combined ensemble performance depends on both the individual system performances and the diversity between the system behaviours, as is mentioned in [61].

Performing recognition without Dropout appears to consistently outperform MBR combination decoding. One possible reason may be that performing recognition without Dropout effectively averages the parameters over all possible Dropout masks, and therefore considers a large ensemble size. On the other hand, only 4 decoding runs were used to generate the MBR combination decoding ensemble. However, it was found that the MBR combination decoding WER saturated at 25.7% as the ensemble size was increased, with a Dropout rate of 0.1. This may suggest that parameter-level combination is more effective for this form of ensemble diversity.

The best performance is obtained using a small Dropout rate of 0.1 and performing recognition without Dropout. However, this is not significantly better than training without Dropout, with a null hypothesis probability of 0.107, and therefore the remaining experiments in this thesis do not train with Dropout.

8.1.3 Topology diversity

Both using multiple random parameter initialisations and Monte Carlo Dropout can generate an ensemble with a diversity of model parameters within a fixed model topology. However, using only a single model topology may limit the diversity of the ensemble. A range of possible acoustic model topologies are discussed in Section 2.2.5. The next experiment investigates combining an ensemble that contains systems with two acoustic model topologies, a feed-forward DNN and a BLSTM, described in Table 7.2. Ensembles of 4 DNNs and 4 BLSTMs were sequence-trained in the AMI-IHM dataset. Within each topology type,

multiple sets of model parameters were obtained by beginning training from different random parameter initialisations. Table 8.3 shows the performances of these ensembles.

Table 8.3 Random initialisation with different acoustic model topologies, in AMI-IHM. 4 models from different random parameter initialisations were generated for each topology. All systems were trained with the $\mathcal{F}_{\text{sMBR}}$ criterion, and were combined with MBR combination decoding. BLSTMs are more diversity, and additional combination gains can be obtained by combining different topologies.

	Single system WER (%)				Combined	
Ensemble	mean	best	worst	std dev	WER (%)	cross-WER (%)
4×DNN	25.7	25.6	25.8	0.10	24.9	11.8
4×BLSTM	25.1	24.8	25.2	0.19	22.1	19.4
4×DNN + 4×BLSTM	25.4	24.8	25.8	0.39	21.9	19.2

The two model topologies are first treated separately, to gain insight into the diversities that can be obtained by using only different sets of model parameters within each topology. From the results, it can be seen that the BLSTM topology has both better individual system performances and diversity between the different model parameter sets. The greater diversity suggests that the BLSTM is a more flexible topology than the feed-forward DNN, and is therefore able to exhibit a wider variety of possible behaviours. The greater BLSTM ensemble diversity leads to a larger relative gain in combination of 12.0%, compared to that of the DNN ensemble of 3.1%.

An ensemble with a diversity of model topologies is constructed by merging the 4 DNNs and 4 BLSTMs into a larger ensemble. This ensemble has a slightly better combined performance than the BLSTM ensemble that only has a diversity of model parameter sets. However, this may not be statistically significant, with a null hypothesis probability of 0.093. The smaller cross-WER of the ensemble with both topologies, compared to that of the BLSTM ensemble, is an artefact of the cross-WER diversity measure defined in (3.64), as it includes contributions from the DNN to DNN model pairs, which have smaller cross-WERs. To obtain a better sense of the diversity between different topologies, the cross-WER averaged over all DNN to BLSTM pairs is 21.9% (not shown in Table 8.3). This is greater than the diversity between BLSTMs alone. The results suggest that it may be possible to obtain more diverse behaviours when using a variety of model topologies, than when using a single model topology.

8.2 Echo state network feature diversity

Besides having a diversity of acoustic models, it is also possible to have a diversity of feature representations. Different forms of hand-crafted features can be used for each member of the ensemble. However, the size of such an ensemble is limited by the variety of hand-crafted features. As is described in Section 3.3.1, multiple different NN feature extractors can instead be used to generate the ensemble. These multiple NN feature extractors can be obtained using the methods discussed in Sections 3.1 and 3.2. However, these approaches can be computationally expensive. In Section 3.5, the ESN is proposed as a method of randomly sampling multiple feature representations in a computationally cheap manner. The feature-level combination method, discussed in Section 3.4.3, can be used to efficiently combine such an ensemble when performing recognition.

This section investigates the effectiveness of generating an ensemble using ESN random feature projections, and combining the ensemble using feature-level combination. ESNs with different projection sizes were randomly sampled. As is discussed in Section 3.5, since the ESN is not trained, it may be important to place constraints on the ESN parameters for stability. The ESN parameters were scaled using (3.62) and (3.63), to have a recurrent matrix spectral radius, (3.58), of $\breve{\varpi} = 0.6$ and a nonlinearity scale, (3.59), of $\breve{\zeta} = 0.5$. These hyper-parameter values were chosen, as preliminary hyper-parameter sweeps showed them to produce reasonable performances. Each ESN represents an ensemble of multiple sampled feature representations. The ESN projection size determines the number of feature representation samples, which in the current context can be interpreted as the ensemble size. These ESN feature samples were combined together using feature-level combination, by training a single feed-forward DNN classifier on the ESN projections. In this preliminary experiment, the DNN classifier was trained using the cross-entropy criterion in (2.80). The AMI-IHM dataset was used.

Table 8.4 Feature-level combination of ESN random feature projections, in AMI-IHM. Feature-level combination was performed by training a DNN acoustic model classifer on each ESN with a different projection dimension, toward the $\mathcal{F}_{\text{CE}}$ criterion. An ESN with feature-level combination does not provide any performance gains.

ESN projection size	WER (%)
2000	28.7
3000	29.0
4000	29.3
5000	29.5
none	28.4

The results in Table 8.4 suggest that using the ESN to sample feature representations, together with feature-level combination, does not provide any ensemble combination gains. Increasing the ensemble size by using a larger ESN projection does not improve the ensemble performance. A larger ESN projection may possibly lead to a better ensemble performance, if the added features express new useful information. One method of estimating the amount of information within the ensemble of feature representations is to measure the rank of the covariance matrix of the features. This expresses the number of feature dimensions that are not linearly correlated. For all projection sizes in Table 8.4, the covariance matrices of the ESN projections were always measured to be full-rank, thereby suggesting that new information is added with each feature dimension. However, this may not be a reliable indicator of new useful information, as it only measures the linear correlation between the features. It may be possible that the projections reside within a low-dimensional nonlinear manifold, that may not increase in dimension as the projection size increases.

8.3 State cluster diversity

As is discussed in Section 3.3.3, ensemble diversity can be obtained by using different sets of state clusters between the systems. The set of state clusters is defined by the phonetic decision tree, according to (2.28). In the AMI-IHM dataset, ensembles with 4 decision trees each were generated using the random forest method, described in Section 3.3.3, where the split at each iteration was sampled uniformly from the best 5. In the AMI-IHM dataset, each of the 43 centre phones had a different tree root[2]. This root tree, with 43 leaves, was the initial tree, from which all random forest trees were built upon.

Before training the systems, it may be useful to assess the diversity that can be obtained by using multiple decision trees. The Tree Cluster Divergence (TCD) of (3.69) provides an approximate measure of such a diversity. This measures the KL-divergence between the distributions of the observations within the leaves of the different decision trees, under the assumption that these observation features are Gaussian distributed. The TCD between the decision trees is larger if the leaves are more different. Figure 8.1 shows the TCD for ensembles with various decision tree sizes. Here, all decision trees within each ensemble were constrained to have the same number of leaves. When the number of leaves is small, there are relatively few possible permutations of how to clusters the logical context-dependent states, leading to a small TCD. The TCD initially rises with the number of leaves, as the number of ways to form clusters increases. In the limit as the number of leaves approaches

[2]Different HMM state indexes were not forced to have different tree roots. The clustering of different HMM state indexes can be learnt from the data.

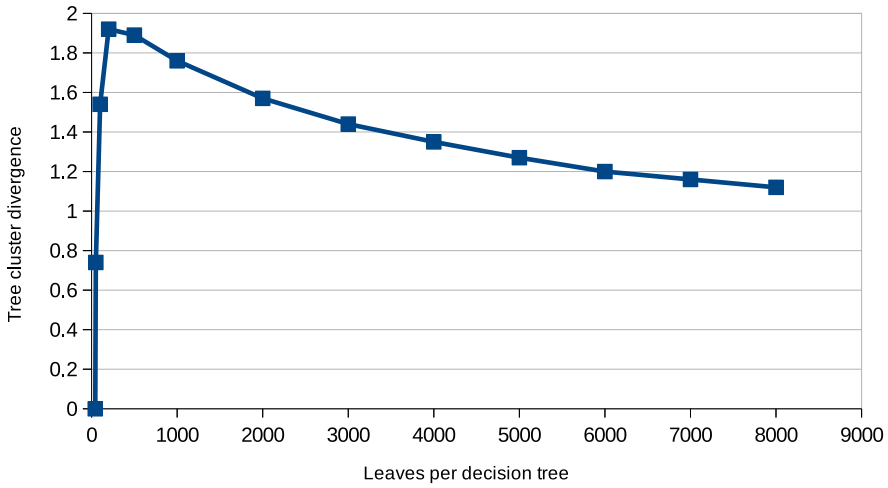


Fig. 8.1 Tree cluster divergence vs decision tree size, in AMI-IHM. 4 random forest decision trees were generated for each decision tree size, by sampling uniformly from the 5-best splits at each training iteration. The trees had 43 root nodes, one for each centre phone.

the total number of logical context-dependent states, the number of permutations of forming clusters decreases. After initially rising, the TCD therefore then gradually decreases as the number of leaves increases. The maximum TCD occurs at around 200 leaves.

Table 8.5 Ensembles with different decision trees of various sizes, in AMI-IHM. Each ensemble had 4 DNNs, each with a different random forest tree, and trained toward the $\mathcal{F}_{\text{sMBR}}$ criterion. MBR combination decoding was used for combination. Increasing the decision tree sizes up to 4000 leaves reduces the diversity but leads to better individual system performances, and therefore better a better combined performance.

No. of decision tree leaves	Single system WER (%) mean	best	worst	std dev	Combined WER (%)	cross-WER (%)
500	28.7	28.7	28.8	0.05	26.8	17.9
2000	26.7	26.6	26.8	0.08	25.2	16.0
4000	26.0	25.9	26.2	0.13	24.5	15.2

A separate DNN acoustic model was used with each decision tree, and trained using the sequence-level $\mathcal{F}_{\text{sMBR}}$ criterion of (2.83). The forced alignment targets for the initial cross-entropy training were obtained from a common set of forced alignments from a GMM-HMM system. These common forced alignments were mapped to each of the different decision trees. Table 8.5 shows the ASR performance of ensembles with decision trees of different sizes. The trends in the measured cross-WER diversity agree with Figure 8.1. The ensemble with small 500-leaves decision trees has the largest cross-WER diversity. However, systems

with smaller decision trees also have worse individual performances. This may be because a smaller decision tree leads to a poorer phonetic resolution. Therefore, although an ensemble with smaller decision trees is more diverse, the worse single system performances lead to a worse combined performance.

Table 8.6 Comparison between ensembles with a diversity of acoustic model parameters only and a diversity of state cluster sets with different decision trees, in AMI-IHM. Each ensemble had 4 DNNs, trained toward the $\mathcal{F}_{\text{sMBR}}$ criterion, and was combined using MBR combination decoding. Having different decision trees leads to a wider diversity and better combination gain.

Dataset	Ensemble diversity	Single system WER (%) mean	best	worst	std dev	Combined WER (%)	cross-WER (%)
207V	parameter	47.8	47.6	48.0	0.18	46.3	21.9
	state cluster	48.3	48.0	48.4	0.17	45.8	28.4
AMI-IHM	parameter	25.7	25.6	25.8	0.10	24.9	11.8
	state cluster	26.0	25.9	26.2	0.13	24.5	15.2
HUB4	parameter	9.2	9.1	9.3	0.10	8.8	5.6
	state cluster	9.3	9.2	9.4	0.10	8.7	7.0

Table 8.6 compares ensembles with a state cluster diversity by using different decision trees, and ensembles with only model parameter diversity by training multiple models with the same decision tree from different random parameter initialisations. The 207V, AMI-IHM and HUB4 datasets were used here, to demonstrate how the ensembles perform over a variety of tasks. These datasets have 3, 81, and 144 hours of training data and used decision tree sizes of 1000, 4000, and 6000 leaves respectively. When less training data is available, smaller decision trees reduce the number of trainable parameters in the acoustic model, thereby allowing for better model generalisation. When more training data is available, using a larger decision tree increases the model's capacity, and may lead to a better performance. The decision trees used in the ensembles with only a diversity of acoustic model parameters were generated using greedy splits.

The results suggest that using different sets of state clusters can yield a wider diversity and better combined performance than only using different sets of model parameters. In AMI-IHM, the ensemble with state cluster diversity has a cross-WER of 15.2% and combined WER of 24.5%, which is better than the ensemble with only model parameter diversity, having a cross-WER of 11.8% and a combined WER of 24.9%. The benefit of using multiple decision trees appears to be greater in situations where the quantity of training data is more limited, and therefore smaller decision trees tend to be used. In the 207V and AMI-IHM datasets, the combined ensembles with different decision trees are able to significantly outperform the

ensembles with only different model parameters, with null hypothesis probabilities of less than 0.001 for both datasets. In HUB4, the performance difference between these ensembles may not be significant, with a null hypothesis probability of 0.358.

The single systems with random forest decision trees perform worse than those that use greedy splits, in the ensembles with only a diversity of acoustic model parameters, possibly because the random forest decision trees are less optimal. In AMI-IHM, the mean single system WER with a random forest tree is 26.0%, which is worse than that with a greedy tree of 25.7%. The random forest method used here is a fairly simple approach of obtaining different decision trees. Other methods investigated in [15, 162] explicitly train the decision trees to be different. It may be possible to obtain greater diversity and combination gains using such methods.

8.4 Sub-word unit diversity

Section 3.3.4 has described the possibility of constructing an ensemble with different sets of sub-word units. ASR systems are often built using a phonetic dictionary. However, the linguistic expertise needed to obtain a phonetic dictionary may be expensive to acquire. An alternative is to use a graphemic dictionary, where words are decomposed as how they are spelt. This eliminates the need for the linguistic expertise required to obtain the dictionary. However, while a phonetic dictionary can contain multiple decompositions of each word, corresponding to different pronunciations, a graphemic dictionary only has a single decomposition of each word. Furthermore, many languages may not have a close grapheme-to-phone relationship. Although this may place a greater demand on the acoustic model of a graphemic system, it can also lead to different behaviours between phonetic and graphemic systems. This experiment considers the combination of a phonetic and a graphemic system, to leverage upon this diversity. The AMI-IHM dataset was used here, which comprises data in the English language. Work in [83] suggests that for English, the grapheme-to-phone relationship may not be very close. As such, phonetic and graphemic English systems may behave significantly differently. A graphemic system was built using the same DNN topology as the phonetic system. The graphemic decision tree was also built to have the same number of 4000 leaves as the phonetic decision tree. Both systems were trained using the sequence-level $\mathcal{F}_{\text{sMBR}}$ criterion of (2.83).

The performances of the phonetic and graphemic systems are shown in Table 8.7, as well as their combined performance. The graphemic system performs worse than the phonetic system, agreeing with the observations in [83, 150]. This suggests that the grapheme-to-phone relationship in English is not very close, and that the accuracy of the dictionary can

Table 8.7 Combination of a phonetic and a graphemic system, in AMI-IHM. The systems had DNN acoustic models, trained toward the $\mathcal{F}_{\text{sMBR}}$ criterion, and were combined using MBR combination decoding. Phonetic and graphemic systems are highly diverse.

Single system WER (%)		Combined	
phonetic	graphemic	WER (%)	cross-WER (%)
25.7	27.2	24.7	19.1

significantly affect the ASR performance. Work in [150] has shown that the performance difference between phonetic and graphemic systems can be reduced by using acoustic models that take into account longer temporal contexts.

Despite the difference between the single system performances, combining together the phonetic and graphemic systems results in a large performance gain. The cross-WER diversity between the systems is larger than those measured for ensembles with only diversities of acoustic model parameters or sets of state clusters, shown in Table 8.6. This suggests that the behaviours of phonetic and graphemic systems are highly diverse. Using different sets of sub-word units also requires using different sets of state clusters, which may contribute to the diversity. However, some contribution to the large cross-WER may be due to the large difference between the phonetic and graphemic system performances.

8.5 Recurrent neural network language model diversity

Another form of diversity that can be incorporated into an ensemble is a diversity of language models, as is discussed in Section 3.3.5. A diversity of different language model topologies can be considered, by combining an n-gram and an RNN language model. Two methods are considered for combining together these language models. First, the language models can be combined by interpolating their probabilities in (3.27), repeated here as

$$P\left(\boldsymbol{\omega}_{1:L}\middle|\widehat{\boldsymbol{\Phi}}\right) = \sum_{m=1}^{M} \lambda_m P\left(\boldsymbol{\omega}_{1:L}|\boldsymbol{\Phi}^m\right), \tag{8.1}$$

where $\boldsymbol{\Phi}^m$ represents each of the different language models, and the interpolation weights satisfy $0 \leq \lambda_m \leq 1$ and $\sum_m \lambda_m = 1$. These interpolated language model probabilities can then be used to compute the hypothesis posteriors using (2.47). Second, hypothesis-level combination can be used. In hypothesis-level MBR combination decoding of (3.29), the hypothesis posteriors of each system are first computed using (2.47), and then combined as a

weighted average,

$$P\left(\boldsymbol{\omega}\middle|\mathbf{O}_{1:T},\widehat{\boldsymbol{\Phi}}\right)=\sum_{m=1}^{M}\lambda_m\frac{P^{\gamma}\left(\boldsymbol{\omega}|\boldsymbol{\Phi}^m\right)\sum\limits_{\mathbf{s}_{1:T}\in\mathcal{G}_{\boldsymbol{\omega}}}\prod\limits_{t=1}^{T}P^{\gamma}\left(s_t|s_{t-1}\right)\mathcal{A}^{\kappa}\left(\mathbf{o}_t,s_t\right)}{\sum\limits_{\boldsymbol{\omega}'}P^{\gamma}\left(\boldsymbol{\omega}'|\boldsymbol{\Phi}^m\right)\sum\limits_{\mathbf{s}'_{1:T}\in\mathcal{G}_{\boldsymbol{\omega}'}}\prod\limits_{t=1}^{T}P^{\gamma}\left(s'_t|s'_{t-1}\right)\mathcal{A}^{\kappa}\left(\mathbf{o}_t,s'_t\right)}. \tag{8.2}$$

These two combination methods are not equivalent, because the hypothesis posteriors for each member of the ensemble are separately normalised in (8.2).

Table 8.8 Combination of 4-gram and RNN Language Models (LM), in AMI-IHM. A single $\mathcal{F}_{\text{sMBR}}$-trained DNN was used as the acoustic model. Equal interpolation weights were used for both LM interpolation and MBR combination decoding. Combining 4-gram and RNN language models yields a performance gain.

Single system WER (%)		Combined WER (%)		
4-gram	RNN	LM interpolation	hypothesis	cross-WER (%)
25.7	25.7	25.3	25.2	11.7

The AMI-IHM dataset is used for this experiment. An RNN language model was trained using the $\mathcal{F}_{\text{LM-ML}}$ criterion in (2.21) on the training data manual transcripts, using a single RNN layer with 100 nodes, in the RNNLM toolkit[3] [103]. In order to limit the number of parameters and the softmax computation time, the RNN language model output layer had separate nodes for only the 50000 most frequent words in the training data, out of a total of 52549 words in the dictionary. All other words were mapped to a single 50001st output node. The input of the RNN language model had the same collection of nodes as the output. The RNN language model, or an interpolation of the RNN and 4-gram language models, was used to rescore the decoding lattices[4] [96], generated from a sequence-trained DNN acoustic model. Table 8.8 shows the results of an ensemble combination when using these different language models. An equal interpolation weight was used between the two language models. Rescoring the lattice using either the 4-gram or RNN language model alone leads to a similar single system performance. The results suggest that diversity and significant performance gains can be obtained by combining a 4-gram and an RNN language model. Hypothesis-level combination may not be significantly better than language model interpolation using (8.1), with a null hypothesis probability of 0.003.

Other than combining together language models with different topologies, it is also possible to combine language models with different sets of model parameters within the

[3]The experiments in this thesis used version 0.4b, which can be found at https://github.com/mspandit/rnnlm. Older versions of the RNNLM toolkit can be found at http://www.fit.vutbr.cz/ imikolov/rnnlm.

[4]Lattice rescoring with the RNN language model is performed in Kaldi.

same topology. Multiple RNN language models with different sets of model parameters can be generated using the methods discussed in Sections 3.1 and 3.2, similarly to generating multiple acoustic models. An ensemble of 4 RNN language models was generated, by beginning training from different random parameter initialisations. All RNN language models used the same topology.

Table 8.9 shows the performances of ensembles with a diversity of RNN language model parameters. Various amounts of 4-gram language model interpolation were used, as Table 8.8 shows that this can improve the performance. The different RNN language models were combined here using hypothesis-level MBR combination decoding of (3.29), with equal interpolation weights.

Table 8.9 Ensemble with multiple RNN language models from different random parameter initialisations, in AMI-IHM. In each ensemble with 4-gram interpolation less than 1.0, the decoding lattice from a single DNN acoustic model was rescored separately 4 times, with separate RNN language models, and combined with MBR combination decoding. RNN language models from different random parameter initialisations do not yield much diversity.

4-gram	Single system WER (%)				Combined	
interpolation	mean	best	worst	std dev	WER (%)	Cross-WER (%)
1.0	25.7	-	-	-	-	-
0.5	25.3	25.2	25.3	0.05	25.2	3.7
0.0	25.7	25.6	25.7	0.06	25.3	6.9

The results suggest that very little diversity is obtained from an ensemble of different sets of RNN language model parameters. When used without 4-gram language model interpolation, the measured cross-WER between the RNN language models is small, less than the cross-WER between a 4-gram and an RNN language model in Table 8.8. However, combining the systems with different RNN language model parameters still leads to a similar performance as combining a 4-gram and an RNN language model, in Table 8.8. Interpolating the 4-gram language model with each of the RNN language models improves the single system performances. However, this further reduces the diversity between the different RNN language models, and leads to no significant gain when combining across the different RNN language model parameter sets. The different language models were used to rescore a common set of decoding lattices. As such, each language model cannot add any new hypotheses, but can only change the probabilities of each existing hypothesis, and thereby change the rank order of the hypotheses. Pruning of the decoding lattices may limit the set of hypotheses contained within them. This may be a contributing factor to the low diversity.

The mean perplexity of the RNN language models measured on the *eval* transcriptions is 64.8 and the standard deviation is 0.32. As compared to these, the perplexity of the 4-gram

language model is 143.9. These perplexity measurements also suggest that there is much more diversity between the RNN and 4-gram language model behaviours than between each of the RNN language models. Similarly to the experiments on acoustic model diversity in Section 8.1, using a diversity of language model topologies also leads to a greater diversity than using just different sets of language model parameters within the same topology.

8.6 Hypothesis and frame-level combination methods

The previous experiments explore ensembles with different forms of diversities. These ensembles have mostly been combined using hypothesis-level MBR combination decoding of (3.29). There are also other methods of performing combination, both at the hypothesis and frame levels, as is discussed in Section 3.4. Hypothesis-level combination is more general than frame-level combination, as it can be used with more forms of ensemble diversities. However, frame-level combination is less computationally expensive, as only a single lattice needs to be processed for the whole ensemble when performing recognition. This section compares the various hypothesis and frame-level combination methods. Ensembles of 4 DNNs were generated by beginning training from different random parameter initialisations. These systems were all sequence-trained. Ensembles were generated in each of the 207V, AMI-IHM, and HUB4 datasets. Table 8.10 shows the results of combining these ensembles using the various hypothesis and frame-level combination methods.

Table 8.10 Ensemble combination methods. Each ensemble had 4 DNNs beginning from different random parameter initialisations, trained toward the $\mathcal{F}_{\text{sMBR}}$ criterion. Equal interpolation weights are used. Hypothesis-level combination outperforms frame-level combination on 207V, when the quantity of training data is small.

	WER (%)		
Combination	HUB4	AMI-IHM	207V
mean single	9.2	25.7	47.8
Hypothesis level			
MBR combination	8.8	24.9	46.3
CNC	8.8	24.8	46.4
ROVER	8.8	24.9	46.8
Frame level			
posterior sum	8.8	24.9	46.7
likelihood sum	8.8	24.9	46.7
product	8.8	25.0	46.8

The results suggest that any difference in the performance of the different combination methods is more evident only when operating at a higher WER. The diversities for the

Table 8.11 Ensemble diversity. Training with less data yields systems with wider hypothesis-level diversity. However, the frame-level diversity with less data is smaller, because smaller decision trees were used.

Diversity measure	HUB4	AMI-IHM	207V
cross-WER (%)	5.6	11.8	21.9
cross-FER (%)	48.4	47.8	43.2

ensembles in each dataset are shown in Table 8.11. The cross-WER diversity results suggest that when operating at higher WERs, the systems in the ensemble are less certain about their classifications, as there are more disagreements between the systems. For 207V, MBR combination decoding and CNC significantly outperform ROVER, with null hypothesis probabilities of less than 0.001 for both comparisons. At the hypothesis level, MBR combination decoding and CNC consider multiple hypotheses from each system, and also their associated scores. On the other hand, the ROVER configuration that was used only considers the 1-best hypothesis from each system. This does not take into account any information about the uncertainty that each system has about its hypothesis classification. Such information may be particularly useful when operating at higher WERs.

The results in Table 8.10 also suggest that hypothesis-level combination performs better than frame-level combination when operating at higher WERs. Hypothesis-level combination operates upon the word or word sequence posteriors, while frame-level combination operates upon the per-frame state cluster posteriors or observation likelihoods. Hypothesis-level combination can therefore leverage upon the greater hypothesis-level diversity that systems have when operating at a higher WER, shown in Table 8.11. On the other hand, frame-level combination can only leverage upon diversity at the frame-level. Table 8.11 shows that although the cross-WER increases with the WER for the different datasets, the frame-level cross-FER, defined in (3.65), decreases. The reason for the decrease in cross-FER is that fewer state cluster classes were used for AMI-IHM, and even fewer for 207V, than for HUB4. The number of state clusters is determined by the number of leaves in the decision trees, which for HUB4, AMI-IHM, and 207V, have 6000, 4000, and 1000 leaves respectively. These different decision tree sizes were chosen to allow for reasonable acoustic modelling generalisation with limited data. Having fewer classes to discriminate between will naturally lead to a reduced cross-FER diversity. Since frame-level combination can only leverage upon diversity at the frame-level, a reduced frame-level diversity leads to less frame-level combination gains.

Although performing recognition using hypothesis-level combination may perform better than frame-level combination, it is more computationally expensive. Table 8.12 compares the

Table 8.12 Recognition times for different combination methods, in AMI-IHM. These times are not strict, as the CPUs upon which they were run were not isolated from other interrupting processes. The times included the contributions from NN forwarding, lattice generation, lattice rescoring, combination, and lattice decoding.

Combination	Real time factor
single system	$\times 2.1$
frame-level likelihood sum	$\times 5.8$
hypothesis-level MBR combination	$\times 8.2$

time required to perform recognition using these combination methods. The times included the contributions from NN forwarding, lattice generation, lattice rescoring, combination, and lattice decoding. Recognition was performed over a cluster of 63 CPUs, and the times were added up over all the CPUs. However, these CPUs were not isolated from other interrupting processes, and as such, the real time factors shown are not strict. Frame-level combination is more computationally expensive than using a single system, as data needs to be fed through each of the separate acoustic models. Hypothesis-level combination is even more computationally expensive, as multiple decoding lattices need to be processed.

8.7 Multi-task ensemble

Although frame-level combination is less computationally expensive than hypothesis-level combination, data still needs to be fed through each of the separate acoustic models. As is discussed in Section 4.2, when using multiple sets of state clusters, the computational cost of performing recognition can be further reduced by merging together the hidden layers across all NN acoustic models, forming the multi-task ensemble. Here, data only needs to be fed through the hidden layers once for the whole ensemble. This section investigates using such a multi-task ensemble, with a diversity of state cluster sets, in the AMI-IHM and 207V datasets. As with experiment in Section 8.3, 4 different decision trees were generated using the random forest method. Multi-task ensembles were trained using the multi-task cross-entropy criterion in (4.5). The multi-task ensembles used the same hidden layer topologies as each of the separate DNN models. A different multi-task output layer was used for each decision tree. The ensembles were combined using frame-level combinations of (3.40) and (4.8), as this is less computationally expensive than hypothesis-level combination. Forced alignments were obtained from a GMM-HMM system for each dataset, and mapped to each of the different decision trees, allowing the time-synchronous state transition assumption of the frame-level combination method, discussed in Section 3.4.2, to be satisfied.

Table 8.13 Multi-task ensemble trained with cross-entropy. 4 random forest decision trees were used, and combination was performed at the frame level. The multi-task ensembles are less diverse.

Dataset	Ensemble	Single system WER (%) mean	best	worst	std dev	Combined WER (%)	cross-WER (%)
207V	separate	50.3	50.0	50.5	0.24	48.4	26.6
	multi-task	50.2	50.0	50.4	0.17	49.4	19.6
AMI-IHM	separate	28.7	28.5	28.9	0.17	27.5	15.9
	multi-task	28.6	28.5	28.7	0.10	27.9	11.9

Table 8.13 compares the performances of the multi-task ensembles, against ensembles of separate acoustic models. The separate systems were trained with the frame-level cross-entropy criterion in (2.80). The results suggest that by using the multi-task cross-entropy criterion, each of the multi-task output layers can achieve a similar performance to each of the separate systems. However, the multi-task ensemble exhibits much less diversity between the behaviours of its output layers, than that between separate systems. This may be caused by the sharing of many of the parameters between the members of the multi-task ensemble. This reduced diversity of the multi-task ensemble leads to smaller WER gains in combination.

However, the multi-task ensemble is less computationally expensive to use than an ensemble of separate systems. The code implementation for the multi-task ensemble combination had not been optimised for computational efficiency. As such, a proper comparison of the recognition times could not be performed. An approximate comparison can be made in terms of the number of model parameters. The ensemble of separate acoustic models in the 207V dataset has 19.9×10^6 parameters, while the multi-task ensemble has only 8.0×10^6 parameters. It is reasonable to expect that having fewer model parameters requires a lower computational cost when performing recognition.

8.7.1 Joint sequence discriminative training

The multi-task ensembles in Table 8.13 had only been trained at the frame level. Work in [86] shows that for a single system, sequence discriminative training can often perform better than frame-level training. In Section 4.2.1, two methods to perform sequence discriminative training of a multi-task ensemble are proposed. The multi-task ensemble can be trained by interpolating together separate sequence discriminative criteria for each output layer, using the criterion of (4.6). When used with a per-frame state-level risk function, this is referred to as $\mathcal{F}_{\text{MT-sMBR}}$. The multi-task ensemble can also be trained by back-propagating the derivative from a single sequence discriminative criterion through a frame-level combination using the

criterion of (4.9). When again used with a per-frame state-level risk function, this is referred to as $\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$.

Table 8.14 Comparison of separate and joint sequence discriminative training of a multi-task ensemble, in AMI-IHM. 4 random forest decision trees were used. Separate sequence training yields a better hypothesis-level combined performance. Joint sequence training has similar performances for both combination methods.

	Single system WER (%)				Combined WER (%)		cross-
Training	mean	best	worst	std dev	frame	hypothesis	WER (%)
Frame level							
cross-entropy, $\mathcal{F}_{\text{MT-CE}}$	28.6	28.5	28.7	0.10	27.9	27.6	11.9
Sequence level							
separate, $\mathcal{F}_{\text{MT-sMBR}}$	25.8	25.7	25.8	0.05	25.3	25.1	10.4
joint, $\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$	26.3	26.1	26.5	0.17	25.4	25.4	12.4

Table 8.14 compares both of these sequence discriminative training methods for a multi-task ensemble, in AMI-IHM. The results show that using both forms of sequence discriminative training outperform frame-level cross-entropy training. As is discussed in Section 4.2.1, separate training using $\mathcal{F}_{\text{MT-sMBR}}$ is related to hypothesis-level combination, while joint training using $\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$ is related to frame-level combination. These relationships are reflected in the results, which show that for separate sequence training, the hypothesis-level combination of the multi-task ensemble with a WER of 25.1% outperforms the frame-level combination with a WER of 25.3%, with a null hypothesis probability of 0.001. When performing joint sequence training, there is no significant difference between the hypothesis and frame-level combined performances. The results also show that the mean single system performance is better when using separate training with a WER of 25.8%, compared to that for joint training with a WER of 26.3%. Separate sequence training optimises each member of the ensemble toward its own criterion. This grounds each ensemble member to perform well on its own, leading to a good single system performance. On the other hand, joint training only optimises the frame-level combined performance of the multi-task ensemble, with no requirement for each individual ensemble member to perform well on its own. As such, the single system performance is not improved as much with joint sequence discriminative training, compared to the improvement from separate sequence discriminative training. However, joint training does train the ensemble members to work well together, which may be a contributing factor to the larger cross-WER diversity of 12.4% after joint training, compared to that for separate training of 10.4%.

Frame-level combination is less computationally expensive than hypothesis-level combination, as only a single decoding lattice needs to be processed for the whole ensemble. When

combining at the frame level, there is no significant performance difference between training separately and jointly, with a null hypothesis probability of 0.126. Furthermore, it is less computationally expensive to perform joint training, as only a single lattice of hypotheses needs to be considered when computing the criterion derivative.

Table 8.15 Sequence discriminative training of separate systems and a multi-task ensemble. Frame-level combination was used. Sequence-trained multi-task ensembles are still not as diverse as separate systems.

Dataset	Ensemble	Training	WER (%) mean single	WER (%) combined	cross-WER (%)
207V	separate	$\mathcal{F}_{\text{CE}}$	50.3	48.4	26.6
		$+\mathcal{F}_{\text{sMBR}}$	48.3	46.0	28.4
	multi-task	$\mathcal{F}_{\text{MT-CE}}$	50.2	49.4	19.6
		$+\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$	48.7	47.8	20.7
AMI-IHM	separate	$\mathcal{F}_{\text{CE}}$	28.7	27.5	15.9
		$+\mathcal{F}_{\text{sMBR}}$	26.0	24.6	15.2
	multi-task	$\mathcal{F}_{\text{MT-CE}}$	28.6	27.9	11.9
		$+\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$	26.3	25.4	12.4

Table 8.15 compares an ensemble of separate sequence-trained systems against a sequence-trained multi-task ensemble. Joint sequence discriminative training was used for the multi-task ensemble. Both ensembles were combined using frame-level combination. The results show that sequence discriminative training is able to improve the performances of both forms of ensembles. However, the multi-task ensembles show less diversity and smaller combination gains, than the ensembles of separate systems. In AMI-IHM, the multi-task ensemble after sequence training has a cross-WER of 12.4% and combined WER of 25.4%, which is worse than the separate systems, with a cross-WER of 15.2% and a combined WER of 24.6%. This lack of diversity may be due to the many parameters that are shared across the different members of the multi-task ensemble.

8.8 Summary

This chapter has investigated various forms of diversities that an ASR ensemble can have. The results suggest that diverse behaviours and significant combination gains can be obtained simply by starting multiple training runs from different random parameter initialisations. Additional diversity can be obtained by using different acoustic model topologies. Using multiple sets of state clusters is also able to provide large ensemble diversity, and seems to be especially beneficial when the quantity of training data is limited and small decision trees are

used. Large ensemble diversity can also be obtained using different sub-word units. Although this chapter has generated ensembles using each of these diversity methods separately, it is possible to concurrently use multiple forms of diversities to obtain an even richer ensemble.

Different hypothesis and frame-level combination methods have been assessed, suggesting that hypothesis-level combination may perform better when operating at higher WERs. However, frame-level combination is less computationally expensive. The computational cost of performing recognition can be further reduced by using a multi-task ensemble. However, the parameter sharing in the multi-task ensemble leads to a reduction in the diversity.

Chapter 9

Experiments on frame-level teacher-student learning

The experiments in Chapter 8 show that significant performance gains can be obtained by combining an ensemble of multiple systems. However, the computational cost of using an ensemble to perform recognition scales linearly with the ensemble size, when using the hypothesis and frame-level combination methods, discussed in Sections 3.4.1 and 3.4.2. This can hinder deployment on devices with limited hardware resources, especially when real-time operation is required. The teacher-student learning method, described in Section 4.3.3, can reduce this computational demand, by training a single student to emulate the combined ensemble. Only this student then needs to be used for recognition.

This chapter investigates frame-level teacher-student learning. Section 9.1 first establishes the ability of a single student to learn from an ensemble of teachers. Section 9.2 aims to determine why the information propagated from the teachers may be useful to the student. Section 9.3 investigates a simple method of incorporating sequence-level information, by performing further sequence training on the student. Section 8.1.1 shows that with standard training, systems are sensitive to the parameter initialisation, and this can be used to obtain diversity. Section 9.4 assess whether teacher-student learning also has a sensitivity to the parameter initialisation.

Ensembles can use different forms of diversities. Section 9.5 assess the ability of students to learn from different model topologies, which Section 8.1.3 shows can yield large diversity. Section 8.3 shows that using different sets of state clusters can also yield large diversity. Section 9.6 assess the methods proposed in Section 5.1 that allow teacher-student learning to be used with this form of diversity.

9.1 Learning from an ensemble of teachers

An ensemble can be compressed into a single student using teacher-student learning, to reduce the computational cost of performing recognition. This initial experiment assesses the ability of a single student to learn from an ensemble of teachers. Students were trained toward ensembles of teachers using the standard frame-level teacher-student learning criterion in (4.15). The DNN teacher ensembles here used a diversity of acoustic model parameters, by beginning multiple training runs from different random parameter initialisations. All teachers were trained using the sequence-level $\mathcal{F}_{\text{sMBR}}$ criterion of (2.83). The students were first pre-trained using the stacked-RBM method for AMI-IHM, and supervised pre-training toward the frame-level teacher-student learning criterion for 207V and HUB4. The students had the same DNN acoustic model topologies as each teacher. These topologies are described in Table 7.2.

Table 9.1 Single DNN systems with frame and sequence-level training.

	WER (%)	
Dataset	$\mathcal{F}_{\text{CE}}$	$+\mathcal{F}_{\text{sMBR}}$
207V	50.2	47.8
AMI-IHM	28.4	25.7
HUB4	10.0	9.2

As a baseline for comparison, Table 9.1 first shows the performances of single systems that were trained using the frame-level cross-entropy criterion in (2.80), and the sequence-level criterion of $\mathcal{F}_{\text{sMBR}}$ in (2.83). Table 9.2 then shows the combined performances of the sequence-trained ensembles, using hypothesis-level MBR combination decoding of (3.29), and the student performances.

Table 9.2 Frame-level teacher-student learning. The ensembles had 4 DNNs from different random parameter initialisations, trained separately with the $\mathcal{F}_{\text{sMBR}}$ criterion. The students had the same DNN topologies as each teacher in the ensembles. The ensembles were combined using MBR combination decoding. Equal interpolation weights were used for both teacher-student learning and ensemble combination. The student is able to more closely approach the combined ensemble performance than can a system trained with $\mathcal{F}_{\text{CE}}$ or $\mathcal{F}_{\text{sMBR}}$.

Dataset	Ensemble WER (%)	Student WER (%)
207V	46.3	46.9
AMI-IHM	24.9	25.1
HUB4	8.8	8.9

The results show that through frame-level teacher-student learning, the student is able to more closely approach the ensemble performance, than through standard frame-level cross-entropy training. This demonstrates that it is possible to compress an ensemble into a single system.

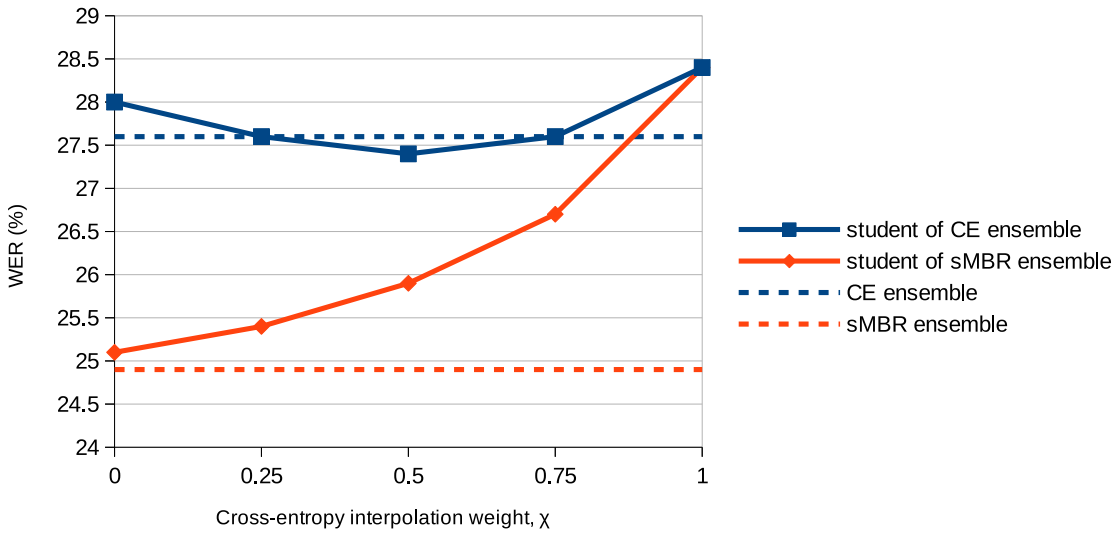


Fig. 9.1 Interpolation of cross-entropy and teacher-student learning criteria, in AMI-IHM. Cross-entropy interpolation helps when training toward the frame-level teachers, but not for the sequence-level teachers.

As is discussed in Section 4.3.3, the cross-entropy and frame-level teacher-student learning criteria are related, as both represent KL-divergences between per-frame state cluster posteriors. The difference between these criteria is in the form of targets. The targets in the cross-entropy criterion are δ-functions at the forced alignments, as in (4.18). On the other hand, the targets in the teacher-student learning criterion are the combined state cluster posteriors of the teacher ensemble of (4.16). The targets from the ensemble are computed without using information about the manual transcriptions. It may be beneficial for the student to learn from both the ensemble and the manual transcriptions. This can be achieved by interpolation the cross-entropy and teacher-student learning criteria together, using (4.20), repeated here as

$$\mathcal{F}(\boldsymbol{\Theta}) = \chi \mathcal{F}_{\text{CE}}(\boldsymbol{\Theta}) + (1-\chi)\,\mathcal{F}_{\text{TS}}^{\text{state}}(\boldsymbol{\Theta})\,. \tag{9.1}$$

Figure 9.1 shows the performances of students trained using different interpolation weights, in AMI-IHM. Two forms of teacher ensembles were used. The teachers in the ensembles were either cross-entropy or sequence-trained. The results suggest that when the teachers have been trained with the frame-level cross-entropy criterion, interpolating

the criteria can lead to a better student performance. The cross-entropy ensemble was trained using the forced alignment targets, and may therefore produce targets that are in good agreement with them. However, for the sequence-trained ensemble, interpolating the cross-entropy criterion when training the student causes the student to gradually back-off toward the performance of a single cross-entropy system.

9.2 Propagating information about classification difficulty

The cross-entropy and frame-level teacher-student learning criteria are related in their forms, with the difference begin in the targets used. As is discussed in Section 4.3.3, one hypothesis of why the teacher posteriors may help the student is because they contain information about how difficult the teachers believe that each frame is to classify. Assuming that an ensemble of teachers has a greater modelling capacity than a single student, then if the ensemble has difficulty in correctly classifying a frame, it may be better for the student to not attempt to produce a low-entropy posterior for that frame. This section presents two experiments to validate this hypothesis.

9.2.1 Form of frame-level targets

In the standard cross-entropy criterion of (4.18), repeated here as

$$\mathcal{F}_{\text{CE}}\left(\boldsymbol{\Theta}\right) - \sum_{t=1}^{T} \sum_{s \in \mathcal{T}} \delta\left(s, s_t^{\text{ref}}\right) \log P\left(s|\mathbf{o}_t, \boldsymbol{\Theta}\right), \tag{9.2}$$

the δ-function targets do not propagate information about how difficult the frames are to classify. These may encourage the trained model to express low-entropy posteriors, whether a frame is difficult to classify or not. The difficulty of classifying each frame can be expressed as disagreements between the classifications of each teacher, and in the entropy of the target posteriors from each individual teacher. Section 4.3.3 proposes several frame-level teacher-student learning criteria, which differ in their forms of targets. The disagreement between the forced alignments from each teacher can be captured using the criterion of (4.22), repeated here as

$$\mathcal{F}_{\text{TS}}^{\text{1best}}\left(\boldsymbol{\Theta}\right) = -\sum_{t=1}^{T} \sum_{s \in \mathcal{T}} \left[\sum_{m=1}^{M} \lambda_m \delta\left(s, s_t^{m,\text{ref}}\right) \right] \log P\left(s|\mathbf{o}_t, \boldsymbol{\Theta}\right), \tag{9.3}$$

where the forced state cluster alignments from each teacher are

$$s_t^{m,\text{ref}} = \arg\max_s P\left(s_t = s \middle| \boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right). \tag{9.4}$$

The targets here take a weighted average of the forced alignments from each teacher. However, this only captures the disagreement between the teachers. Information about how certain each teacher is about its state cluster alignment can be captured by considering the soft alignment distributions, using the criterion of (4.23), repeated here as

$$\mathcal{F}_{\text{TS}}^{\text{soft-align}}\left(\boldsymbol{\Theta}\right) = -\sum_{t=1}^{T}\sum_{s\in\mathcal{T}}\left[\sum_{m=1}^{M}\lambda_m P\left(s_t = s\middle|\boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right)\right]\log P\left(s|\mathbf{o}_t, \boldsymbol{\Theta}\right). \quad (9.5)$$

However, the soft alignment targets utilise information about the manual transcriptions, which the student does not have access to when producing its state cluster posteriors. As such, these targets may by overly certain about the state cluster classifications, expressed in the target posterior entropy. The standard teacher-student learning criterion of $\mathcal{F}_{\text{TS}}^{\text{state}}$ in (4.15) uses the state cluster posteriors of the teachers as the targets, which do not utilise information about the manual transcriptions, and may be better matched with what the student can produce. This criterion is repeated here as

$$\mathcal{F}_{\text{TS}}^{\text{state}}\left(\boldsymbol{\Theta}\right) = -\sum_{t=1}^{T}\sum_{s\in\mathcal{T}}\left[\sum_{m=1}^{M}\lambda_m P\left(s|\mathbf{o}_t, \boldsymbol{\Phi}^m\right)\right]\log P\left(s|\mathbf{o}_t, \boldsymbol{\Theta}\right). \quad (9.6)$$

This experiment assesses the performance of students trained using these criteria, in AMI-IHM. An ensemble of 4 sequence-trained DNNs was generated, by beginning multiple training runs from different random parameter initialisations. Table 9.3 shows the performances of students trained using the various different forms of frame-level teacher-student learning criteria. The baseline cross-entropy system using (9.2), on the first row, was trained toward forced alignment targets from the combined ensemble,

$$s_t^{\text{ref}} = \arg\max_s \sum_{m=1}^{M}\lambda_m P\left(s_t = s\middle|\boldsymbol{\omega}^{\text{ref}}, \mathbf{O}_{1:T}, \boldsymbol{\Phi}^m\right). \quad (9.7)$$

This ensured that the same teachers were used to train the students. As a reference, the combined ensemble WER is 24.9%.

The results suggest that propagating over information about the difficulty of classifying each frame improves the student's performance. The best student is obtained using the standard form of teacher posterior targets, without any information about the manual transcriptions. This form of teacher targets may capture information about the difficulty of classifying each frame that is better matched with what the student can produce.

The student trained with the averaged 1-best forced alignment targets from each teacher outperforms that trained with the cross-entropy criterion, which only uses the single 1-

Table 9.3 Type of targets for frame-level teacher-student learning, in AMI-IHM. All targets were obtained from an ensemble of 4 DNNs from different random parameter initialisations, trained with the $\mathcal{F}_{\text{sMBR}}$ criterion. Using the standard teacher-student learning criterion performs best.

Target	Target entropy	Student WER (%)
forced alignments, $\mathcal{F}_{\text{CE}}$	0.00	27.5
averaged forced alignments, $\mathcal{F}_{\text{TS}}^{\text{1best}}$	0.16	27.2
soft alignments, $\mathcal{F}_{\text{TS}}^{\text{soft-align}}$	0.28	27.2
frame posteriors, $\mathcal{F}_{\text{TS}}^{\text{state}}$	1.63	25.1

best forced alignment from the combined ensemble. However, this difference may not be statistically significant, with a null hypothesis probability of 0.010. These targets capture the disagreement between the forced alignments of the different teachers. Through sequence training, the teachers' behaviours may be allowed to diverge from the common alignment used for the initial cross-entropy training of the teachers. These deviations of the teachers' behaviours may allow the targets here to capture some degree of how difficult the frames are to classify.

Using the soft alignment targets conveys additional information about how difficult each individual teacher believes that each frame is to classify. However, this added information does not appear to result in any significant gain in the student performance.

The results suggest that the standard form of state cluster posterior targets propagate more useful information to the student. These richer targets capture the disagreement between the teachers and the certainty of each teacher about its classification, while not relying on information about the manual transcriptions. These results support the hypothesis that the information about the difficulty of classifying each frame, captured in the teacher targets, may be a reason why the student is able to perform better than a single system that is trained with the standard cross-entropy criterion.

9.2.2 Which frames are most beneficial to the student

This section presents another complementary verification of the hypothesis that it is the information about the difficulty of classifying each frame, propagated from the teachers, that is beneficial for the student. In the experiment in Section 9.2.1, each student was trained using the same form of targets for all frames in the training data. If the hypothesis that information about classification difficulty is helpful to the student is correct, then it can be expected that the teachers' information from frames that are harder to classify will be more useful than information from frames that are easier to classify. The results in [27] support

Table 9.4 Frame categories, based on how frames are classified by the teachers.

Type of frames	Category index
all correct	1
majority correct	2
at least one correct, but no majority	3
all wrong, all disagree	4
all wrong, some agree	5
all wrong, all agree	6

this, by showing that when using a single teacher, the frames that are incorrectly classified by the teacher propagate more useful information to the student. When using an ensemble of teachers, more information can be obtained about the difficulty of classifying each frame, than just whether a single teacher has classified it correctly. One measure of how difficult a frame is to classify is based on the number of teachers in the ensemble that correctly classify it, and how many teachers agree with each other on their classifications. In this experiment, the frames in the training data are divided into the categories shown in Table 9.4, which can approximately be viewed as different levels of classification difficulty. These categories may not be the most appropriate representation of the different degrees of classification difficulty, but it is hoped that they can at least provide some insight into the properties of teacher-student learning.

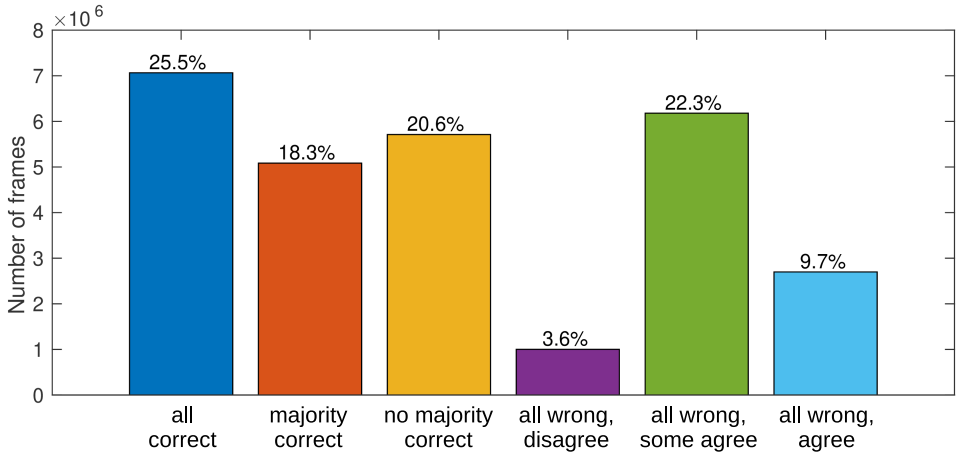


Fig. 9.2 Frame categories of a random initialisation ensemble, in AMI-IHM. The ensemble had 4 DNNs, trained toward the $\mathcal{F}_{\text{sMBR}}$ criterion.

Figure 9.2 shows the distribution of the frames in the AMI-IHM training data among the categories, using an ensemble of sequence-trained DNNs, that had been trained beginning from different random parameter initialisations. The ground truths for the frame classifi-

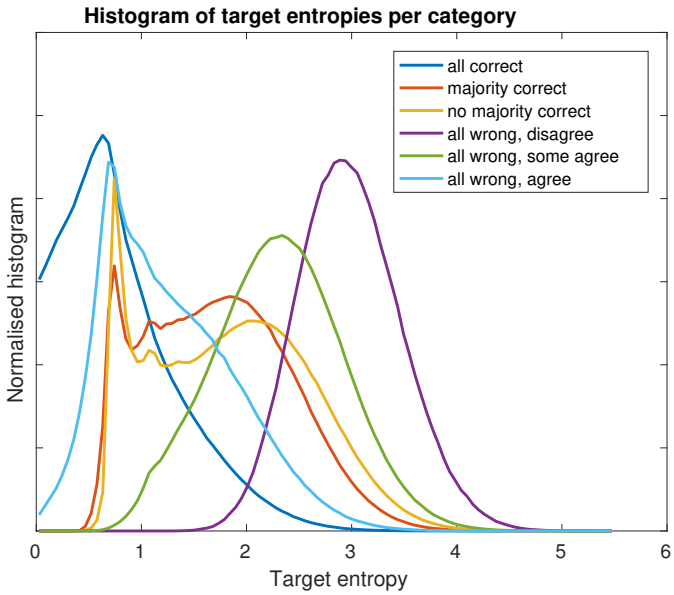


Fig. 9.3 Histograms of entropies of the combined target posteriors for each category, in AMI-IHM.

cations were taken as the forced alignments from a GMM-HMM system. These were the targets that were initially used as the ground truth when performing cross-entropy training of the DNNs. However, further sequence discriminative training of the DNNs may allow their behaviours to diverge from the cross-entropy targets. The distribution of frames between the categories is fairly typical of what has also been observed in other datasets. Figure 9.3 shows the distributions of the combined target entropies for each of the categories. Going from categories 1 to 5, the target entropies generally increase, indicating that the teachers are less certain about their classification of the frames. Category 6 may represent frames where the forced alignment reference may not be the most appropriate, because of the different behaviours of GMM and NN acoustic models. Using forced alignments from a sequence-trained DNN system as the ground truth may yield a reduction in the number of frames in category 6.

Multiple students were trained such that the target used for each frame was either the forced alignment hard target or the soft target from the combined ensemble. Starting from using all forced alignment hard targets, the categories of frames were cumulatively replaced by ensemble soft targets, either from categories 1 to 6 or from 6 to 1, to train a different student. These student performances are shown in Figure 9.4. Each point in the figure represents a student that had been separately trained, beginning from a stacked-RBM parameter initialisation. The results show distinctly different trends between cumulatively replacing the hard targets with teacher posteriors from categories 1 to 6 and from 6 to 1.

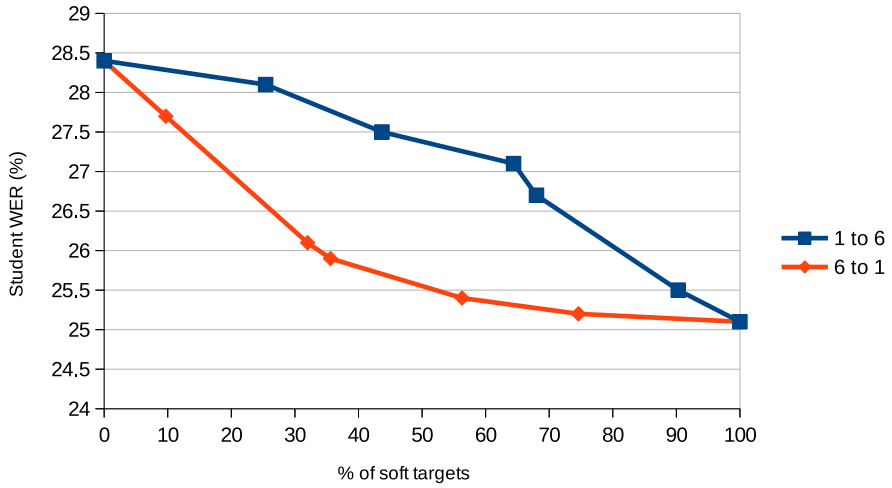


Fig. 9.4 Cumulatively replacing hard targets with soft teacher posteriors, in AMI-IHM. Each point is a separate student, trained from stacked-RBM initialisation, using a different set of hard and soft targets. Teacher posteriors for the frames that are more difficult to classify are more beneficial for the student.

This suggests that the categories of frames that are harder to classify propagate more useful information to the student. This therefore provides additional support to the hypothesis that it is the propagation of information about the difficulty of classifying each frame in the targets that allows the student to perform better than a system trained with the standard cross-entropy criterion, that uses only forced alignment hard targets.

9.3 Incorporating sequence information in the student

The students in the previous experiments were trained using frame-level teacher-student learning criteria. As opposed to this, the teachers in the ensemble were trained using a sequence discriminative criterion. Propagating only frame-level information may not effectively convey the sequence-level behaviours of the teachers. Furthermore, work in [86] suggests that a system trained with a sequence-level criterion can often outperform one trained with a frame-level criterion. Several methods of propagating sequence-level information from the teachers to the student are proposed in Section 6.2. These methods are assessed in Section 10.2. A simpler method to incorporate sequence-level information into the student is to first perform frame-level teacher-student learning using the $\mathcal{F}_{\text{TS}}^{\text{state}}$ criterion in (4.15), then perform further sequence discriminative training, using the student as the model parameter initialisation. Here, the $\mathcal{F}_{\text{sMBR}}$ criterion in (2.83) was used to perform sequence discriminative training. The frame-level DNN students were trained toward the sequence-

trained DNN ensembles, generated by beginning multiple training runs from different random parameter initialisations. The performances of these students are shown in Table 9.5.

Table 9.5 Further sequence discriminative training of the frame-level student. Each ensemble had 4 DNNs, trained with the $\mathcal{F}_{\text{sMBR}}$ criterion, and was combined using MBR combination decoding. The students had the same DNN topologies as each teacher in the ensembles. Further sequence training yields small gains for the students.

Dataset	Combined ensemble WER (%)	Student WER (%) $\mathcal{F}_{\text{TS}}^{\text{state}}$	$+\mathcal{F}_{\text{sMBR}}$
207V	46.3	46.9	46.6
AMI-IHM	24.9	25.1	24.8
HUB4	8.8	8.9	8.8

The results suggest that performing further sequence discriminative training of the student can bring its performance closer to that of the combined ensemble. Although the gains from sequence discriminative training of the students are small, they are consistent across the multiple datasets. The small gains may be because the frame-level information propagated may already contain some information about the sequence-level behaviours of the sequence-trained teachers. However, the gains obtained by performing further sequence discriminative training of the student may indicate that not all information about the sequence-level behaviours of the teachers is effectively propagated by the frame-level state cluster posterior targets.

Single systems that had been sequence-trained after initial cross-entropy training, have WERs of 47.8, 25.7, and 9.2% for the 207V, AMI-IHM, and HUB4 datasets respectively, as is shown in Table 9.1. Performing further sequence discriminative training on the students outperform these systems. This may indicate that the frame-level students are better parameter initialisations for further sequence discriminative training, than systems trained with the standard cross-entropy criterion.

9.4 Diversity of students

Section 8.1.1 shows that the acoustic models are sensitive to the parameter initialisation, when using standard training methods. This sensitivity can be used to obtain diversity, leading to ensemble combination gains. This can also yield disagreements between the classifications of each ensemble member, which can provide useful information to train a student, as is shown in Section 9.2. The experiment in Section 9.3 shows that a frame-level student is a good initialisation for further sequence discriminative training. This may suggest that frame-level teacher-student learning reduces the sensitivity of the model to the initialisation.

This section investigates the sensitivity of the student to the parameter initialisation. To assess this, multiple DNN students were trained toward the same ensemble of sequence-trained DNN teachers, with each student starting from a different random parameter initialisation, in AMI-IHM. The teacher ensemble was also generated by beginning multiple training runs from different random parameter initialisations. The multiple students were used to form an ensemble of students, to assess the differences between their behaviours, and thus to ascertain the sensitivity of the student to the parameter initialisation. Further sequence discriminative training was also performed on the students. These results are shown in Table 9.6.

Table 9.6 Ensemble of multiple students from different random initialisations, in AMI-IHM. The students were all trained toward the same ensemble of teachers. There is only a small diversity between the students.

		Single system WER (%)				Combined	
Ensemble	Training	mean	best	worst	std dev	WER (%)	cross-WER (%)
Teachers	$\mathcal{F}_{\text{CE}}$	28.4	28.3	28.5	0.08	27.6	11.4
	$+\mathcal{F}_{\text{sMBR}}$	25.7	25.6	25.8	0.10	24.9	11.8
Students	$\mathcal{F}^{\text{state}}_{\text{TS}}$	25.1	25.0	25.1	0.05	25.0	4.5
	$+\mathcal{F}_{\text{sMBR}}$	24.7	24.5	24.8	0.13	24.4	7.0

The results show that the diversity between the students, with a cross-WER of 4.5%, is much less than between systems that were trained with the standard cross-entropy and sequence discriminative criteria, with cross-WERs of 11.4% and 11.8% respectively. This suggests that when using frame-level teacher-student learning, the student behaviour is fairly insensitive to the parameter initialisation. This may be because the soft posterior targets from the teacher ensemble are easier to learn than cross-entropy forced alignment hard targets. This may yield a simpler error surface to optimise the student's parameters over. The lack of diversity between the frame-level students leads to there being no significant gain through combination. Further sequence discriminative training of the students appears to increase the diversity between the students, from a cross-WER of 4.5% to 7.0%. This may be because separate sequence discriminative training may allow the students' behaviours to diverge away from what was learnt from the teachers' posteriors. However, despite this larger diversity between the students with further sequence discriminative training, their combined WER of 24.4% is still not significantly better than the best single student WER of 24.5%, with a null hypothesis probability of 0.042.

Despite this lack of significant gains when performing combination over the students, the combined students after further sequence discriminative training, with a WER of 24.4%, performs better than the combined teachers, having a WER of 24.9%, with a null hypothesis probability less than 0.001. It is interesting to consider using this ensemble of students as

teachers to train a second generation of students. This is shown in Table 9.7. Here, the first generation of students, which learn from the original teacher ensemble, are referred to as TS1. The second generation of students, which learn from the first generation after further sequence discriminative training, are referred to as TS2.

Table 9.7 Training students toward an ensemble of students, in AMI-IHM. 4 students in TS1 began from different random parameter initialisations and were trained toward the $\mathcal{F}_{\text{CE}} + \mathcal{F}_{\text{sMBR}}$ ensemble of teachers. 4 students in TS2 began from different random parameter initialisations and were trained toward the TS1 $+ \mathcal{F}_{\text{sMBR}}$ ensemble of students. The second generation of students have even less diversity.

Teacher ensemble	Student ensemble	Single student WER (%) mean	best	worst	std dev	Combined student WER (%)	cross-WER (%)
-	$\mathcal{F}_{\text{CE}} + \mathcal{F}_{\text{sMBR}}$	25.7	25.6	25.8	0.10	24.9	11.8
$\mathcal{F}_{\text{CE}} + \mathcal{F}_{\text{sMBR}}$	TS1	25.1	25.0	25.1	0.05	25.0	4.5
	$+\mathcal{F}_{\text{sMBR}}$	24.7	24.5	24.8	0.13	24.4	7.0
TS1 $+ \mathcal{F}_{\text{sMBR}}$	TS2	24.6	24.6	24.7	0.05	24.6	3.8
	$+\mathcal{F}_{\text{sMBR}}$	24.8	24.7	24.8	0.06	24.5	5.3

The results suggest that training toward the first generation of students results in better performances for each of the students in the second generation. The mean WER of the first generation is 25.1%, while that of the second generation is 24.6%. However, the diversity between these second generation students is further reduced, from a first generation cross-WER of 4.5% to a second generation cross-WER of 3.8%, suggesting that there is even less sensitivity to the parameter initialisation in this second generation. This low diversity leads to no significant combination gains in the second generation. Also, further sequence discriminative training on the second generation of students does not yield any significant performance improvements.

9.5 Learning from different model topologies

In the previous experiments, the ensembles have only utilised a diversity of acoustic model parameters and the student and teachers have all used the same acoustic model topologies. However, the experiment in Section 8.1.3 shows that additional diversity and combination gains can be obtained by using different acoustic model topologies in the ensemble. For a student to instead learn from an ensemble with a diversity of acoustic model topologies, the student will need to learn from teachers that have different topologies from its own. Different model topologies have different modelling capacities. It is interesting to question how well a

student can learn from teachers that use different topologies, and what constraints there need to be on the student topology to allow it to effectively learn from the teachers.

Another benefit of the ability to learn from different model topologies is to reduce the computational cost of using a recurrent model. Work in [145] suggests that an acoustic model with an RNN topology can outperform a feed-forward model, because of the ability to capture longer span temporal dependencies in the data. However, it can be computationally expensive to use recurrent acoustic models to perform recognition, because unlike feed-forward models, the data needs to be processed sequentially, making it difficult to parallelise the computation. If a feed-forward model can learn to behave like a recurrent model, then the feed-forward model can be used instead, allowing the computation to be easily parallelised. Work in [21] shows that it is possible for a feed-forward DNN student to learn from a single teacher with a recurrent topology. This section expands upon this work, by investigating how well a student can learn from teachers with a different topology from its own, as well as from teachers with a diversity of topologies. The DNN and BLSTM acoustic model topologies, described in Table 7.2, are considered in this section.

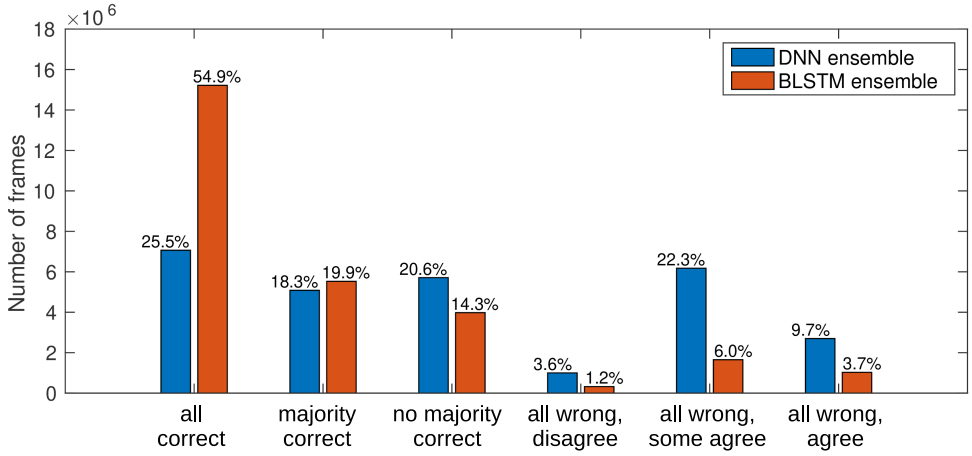


Fig. 9.5 Frame categories of DNN and BLSTM ensembles, in AMI-IHM. For each topology, 4 acoustic models were trained from different random parameter initialisations, toward the $\mathcal{F}_{\text{sMBR}}$ criterion. BLSTMs are able to classify more frames correctly than DNNs.

It is useful to compare the characteristics of the target posteriors generated by feed-forward DNN and recurrent BLSTM teacher ensembles. For each acoustic model topology, an ensemble of 4 sequence-trained systems was generated by beginning training from different random parameter initialisations, in AMI-IHM. Figure 9.5 shows the categorisation of frames according to the classifications of these DNN and BLSTM ensembles. The frame categorisations suggest that the systems in the BLSTM ensemble are able to correctly classify many more frames than those in the DNN ensemble. As such the BLSTM ensemble may

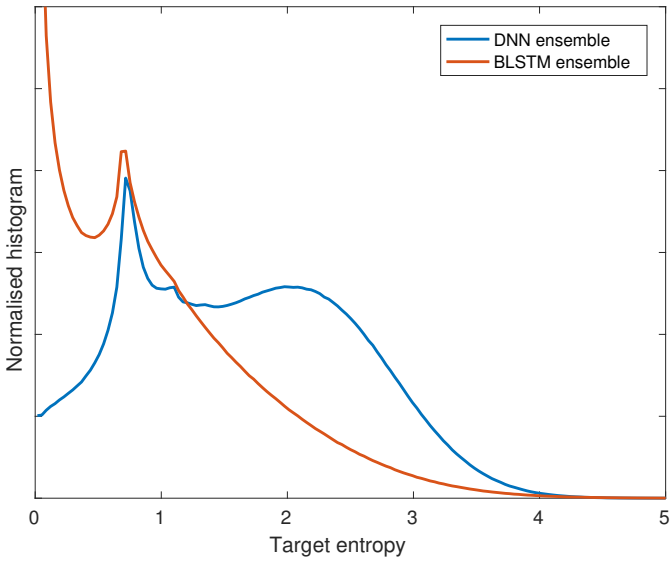


Fig. 9.6 Histograms of entropies of the combined target posteriors from DNN and BLSTM ensembles, in AMI-IHM. The BLSTM ensemble exhibits lower combined posterior entropies.

indicate that the frames are generally less difficult to classify, compared to what the DNN ensemble indicates. Figure 9.6 shows the distribution of entropies in the combined posterior targets generated by the DNN and BLSTM ensembles. The targets from the BLSTM ensemble tend to exhibit much lower entropies than those of the DNN ensemble, showing that the BLSTMs believe that there is less uncertainty about the classification of many frames. The BLSTMs may be able to be more certain about their frame classifications, because they have access to a longer time span of information than the DNNs.

Table 9.8 Feed-forward and recurrent topologies for the teachers and student, in AMI-IHM. Each ensemble had 4 acoustic models from different random parameter initialisations, trained toward the $\mathcal{F}_{\text{sMBR}}$ criterion, and was combined using MBR combination decoding. The students learn best from teachers with the same topology.

Teacher ensemble	Ensemble WER (%)	DNN student WER (%)		BLSTM student WER (%)	
		$\mathcal{F}_{\text{TS}}^{\text{state}}$	$+\mathcal{F}_{\text{sMBR}}$	$\mathcal{F}_{\text{TS}}^{\text{state}}$	$+\mathcal{F}_{\text{sMBR}}$
DNN	24.9	25.1	24.8	24.6	23.6
BLSTM	22.1	26.5	24.3	22.4	22.5

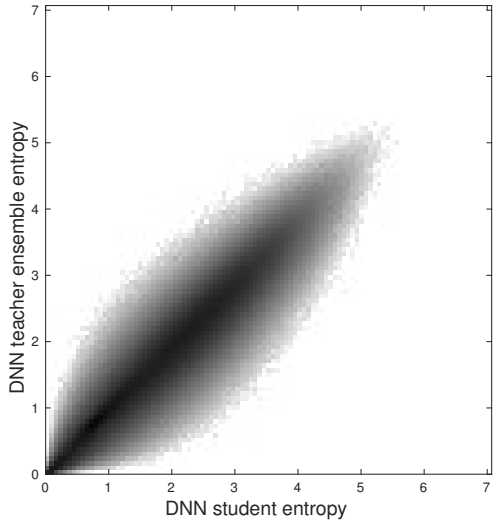
The next experiment assesses the ability of a student to learn from teachers with a different topology than its own. Students with either a DNN or BLSTM topology were trained toward either the DNN or BLSTM teacher ensembles, using frame-level teacher-student learning. As a reference, the mean single system WERs of the teachers are 25.7% and 25.1% for the

DNN and BLSTM ensembles respectively, from Table 8.3. The results in Table 9.8 show that the best student performance, without further sequence discriminative training, for each student topology can be obtained when the student and teacher topologies match. After frame-level teacher-student learning, the DNN student performs better when learning from the DNN ensemble, with a WER of 25.1%, rather than the BLSTM ensemble, with a WER of 26.5%. This is despite the BLSTM ensemble having a better WER of 22.1% than that of the DNN ensemble of 24.9%. The recurrency in the BLSTM teachers allows them to model long span temporal dependencies. With only access to a limited temporal span of information, the DNN student may have difficulty emulating the BLSTM teachers' behaviours. The BLSTM student is able to leverage upon the better performance of the BLSTM ensemble than the DNN ensemble.

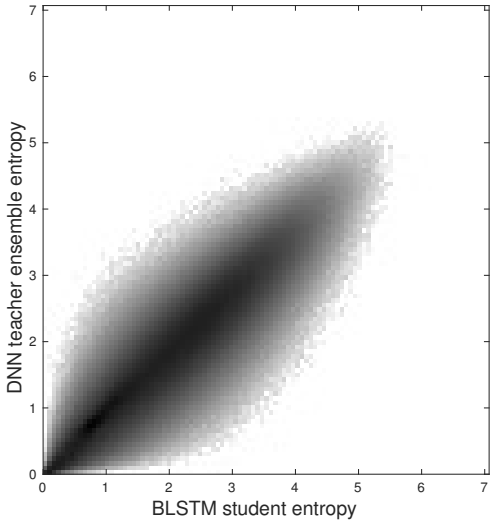
The results in Table 9.8 also suggests that no significant gains can be obtained from performing further sequence discriminative training on the BLSTM student that learns from the BLSTM ensemble. This may indicate that the flexibility of the BLSTM topology allows the BLSTM student to effectively learn to emulate the sequence-trained BLSTM teachers' sequence-level behaviours, even though only frame-level information was propagated over.

As a further investigation into the ability of students to learn from teachers with different topologies, Figure 9.7 shows log-histograms of the per-frame entropies produced by the combined teacher ensembles and their corresponding students, where a darker colour indicates more counts. If the student is able to effectively emulate the combined ensemble behaviour, then it is expected that the histogram will be concentrated along the diagonal. Frames located at the lower right of the plot indicate that the ensemble is more certain about its classification than the student. Frames at the upper left of the plot indicate that the student is more certain about its classification than the teachers. The figures suggest that a BLSTM student is able to effectively emulate both teacher topologies, and a DNN student is able to effectively emulate DNN teachers. However, Figure 9.7c shows that a DNN student is not able to effectively emulate the behaviour of BLSTM teachers. In Figure 9.7c, the mass of the histogram is shifted to the lower right. This indicates that for many frames, the combined BLSTM ensemble is more certain about its classifications, by begin able to express a lower entropy posterior than the DNN student. This agrees with Table 9.8, in suggesting that the DNN student may not be able to effectively emulate the behaviours of BLSTM teachers. These results indicate that it may be important to carefully consider the topology of the student, based on the topologies of the teachers, to allow the student to effectively emulate the teachers' behaviours.

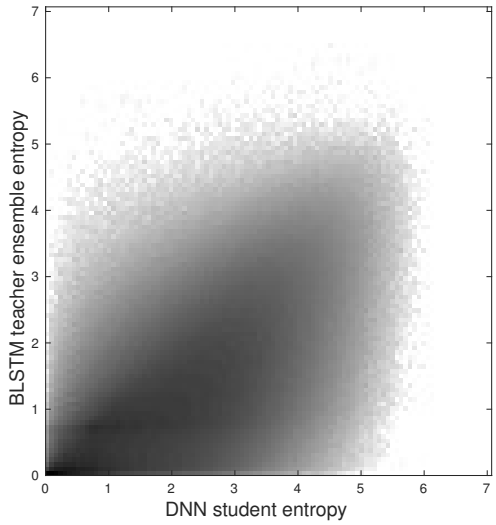
The experiment in Section 8.1.3 suggests that it can be beneficial to use multiple acoustic model topologies within an ensemble. The next experiment assesses how well students



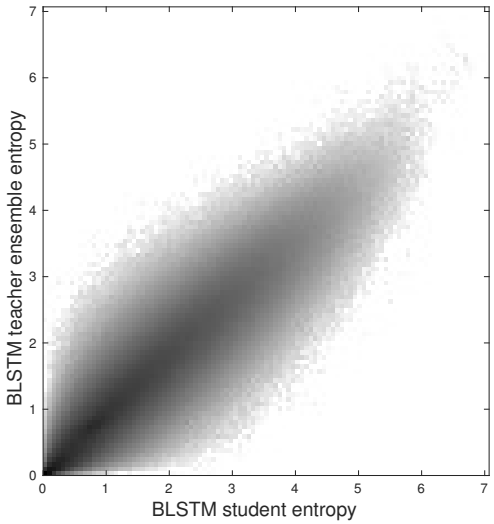
(a) DNN student of DNN ensemble.



(b) BLSTM student of DNN ensemble.



(c) DNN student of BLSTM ensemble.



(d) BLSTM student of BLSTM ensemble.

Fig. 9.7 Log-histograms of per-frame entropies of the ensemble targets and student posteriors. The DNN student is not able to learn effectively from the BLSTM ensemble.

can learn from these multiple topologies together. An ensemble was constructed using all 4 DNNs and 4 BLSTMs together. DNN and BLSTM students were trained toward this ensemble. The results are shown in Table 9.9. As a reference, the combined ensemble WER is 21.9% from Table 8.3.

The results suggest that after frame-level teacher-student learning, the student does not gain much from the added topology diversity, beyond what it can already learn from teachers

Table 9.9 Learning from multiple topologies together, in AMI-IHM. Each student was trained toward an ensemble of 4 DNNs together with 4 BLSTMs, each beginning from a different random parameter initialisation. Learning from multiple topologies does not improve the student performance, but represents a better initialisation for further sequence training.

	Student WER (%)	
Student	$\mathcal{F}_{\text{TS}}^{\text{state}}$	$+\mathcal{F}_{\text{sMBR}}$
DNN	25.1	24.0
BLSTM	22.7	22.1

with its own topology in Table 9.8. However, learning from multiple topologies seems to produce a frame-level student that is a better parameter initialisation for further sequence discriminative training.

9.6 Learning from different sets of state clusters

The previous experiments have demonstrated how teacher-student learning can be used to compress ensembles with diversities of acoustic model parameters and topologies. Another form of diversity that an ensemble can have is for the systems to use different sets of state clusters, as is discussed in Section 3.3.3. The results in Section 8.3 suggest that this can yield significant diversity and combination gains, especially when the quantity of training data is little and the decision trees are small. However, as with other diversity methods, such an ensemble can be computationally expensive to use to perform recognition. It is therefore useful to be able to compress the ensemble using teacher-student learning, as is done in the previous experiments. However, as is explained in Section 5.1, the standard frame-level teacher-student learning criterion in (4.15) requires that all systems use the same set of state clusters. Therefore, standard teacher-student learning cannot be used to compress an ensemble that has a diversity of state cluster sets.

In Section 5.1, two method are proposed to compress such an ensemble. The first method, discussed in Section 5.1.1, aims to map the state cluster posteriors between different decision trees, thereby allowing frame-level posterior information to be propagated from teachers with different sets of state clusters. The second method relates to the multi-task ensemble, discussed in Section 4.2, which compresses an ensemble with a diversity of state cluster sets, by merging together the hidden layers of the NN acoustic models. The experiments in Section 8.7 show that although the multi-task ensemble can be used for compression, it leads to a loss in diversity and combination gains. Section 5.1.2 proposes to overcome this limitation by integrating teacher-student learning with the multi-task ensemble method. This

may allow the multi-task ensemble to learn from the diverse behaviours of separate systems. These two methods are assessed in this current section.

9.6.1 Teacher-student learning across different sets of state clusters

In Section 5.1.1, an extension to frame-level teacher-student learning is proposed to allow for the propagation of frame-level posterior information between teachers and a student that have different sets of state clusters. This trains the student by minimising the KL-divergence between per-frame posteriors of logical context-dependent states, instead of state clusters, using the criterion of $\mathcal{F}_{\text{TS}}^{\text{CD}\prime}$ in (5.5). In this criterion, the target posteriors from each teacher are mapped to the student's set of state clusters, using (5.14). This section investigates the ability of the proposed criterion to compress an ensemble with a diversity of state cluster sets. The performance of this proposed method is compared with standard frame-level teacher-student learning using the criterion of $\mathcal{F}_{\text{TS}}^{\text{state}}$ in (4.15), used to compress an ensemble that does not have a diversity of state cluster sets.

Ensembles of 4 sequence-trained DNNs were generated, in the AMI-IHM and 207V datasets. These ensembles used either a diversity of acoustic model parameters, by beginning multiple training runs from different random parameter initialisations, or a diversity of state cluster sets, by using different decision trees, generated using the random forest method, identically to Section 8.3. Students were trained toward these ensembles. The students had the same DNN topologies as each of the teachers, but used decision trees that were trained using greedy splits. These students had decision trees with 4000 leaves for AMI-IHM and 1000 leaves for 207V, similarly to each of the teachers. As a reference, the combined WERs of the ensembles with model parameter and state cluster diversities are respectively 24.9% and 24.5% for AMI-IHM, and 46.3% and 45.8% for 207V, from Table 8.6.

Table 9.10 shows the performances of these students. The results suggest that by using the proposed criterion, the students can be trained to emulate the ensembles with a diversity of state cluster sets, coming closer to the combined ensemble performances than when using standard cross-entropy and sequence discriminative training. Further sequence discriminative training of the students brings further gains. However, after only frame-level teacher-student learning, the students of the ensembles with state cluster diversity perform worse than those of the ensembles with model parameter diversity, even though the combined ensembles with state cluster diversity outperform the combined ensembles with model parameter diversity. Further sequence discriminative training of the students appears to overcome this performance degradation.

A closer investigation into the nature of the ensemble targets can be made to determine the possible causes of the degraded student performance. Figure 9.8 shows how the training data

Table 9.10 Frame-level teacher-student learning with different sets of state clusters. Each ensemble had 4 $\mathcal{F}_{\text{sMBR}}$-trained DNNs. The students used the same DNN topology as each teacher, but with the greedy decision tree. The student is able to come closer to the combined performance of the ensemble with state cluster diversity, than can standard $\mathcal{F}_{\text{CE}}$ and $\mathcal{F}_{\text{sMBR}}$ systems.

Training	Ensemble diversity	WER (%) AMI-IHM	207V
$\mathcal{F}_{\text{CE}}$	-	28.4	50.2
$+\mathcal{F}_{\text{sMBR}}$		25.7	47.8
$\mathcal{F}_{\text{TS}}^{\text{state}}$	parameters	25.1	46.9
$+\mathcal{F}_{\text{sMBR}}$		24.8	46.6
$\mathcal{F}_{\text{TS}}^{\text{CD}'}$	state clusters	25.5	47.3
$+\mathcal{F}_{\text{sMBR}}$		24.6	46.6

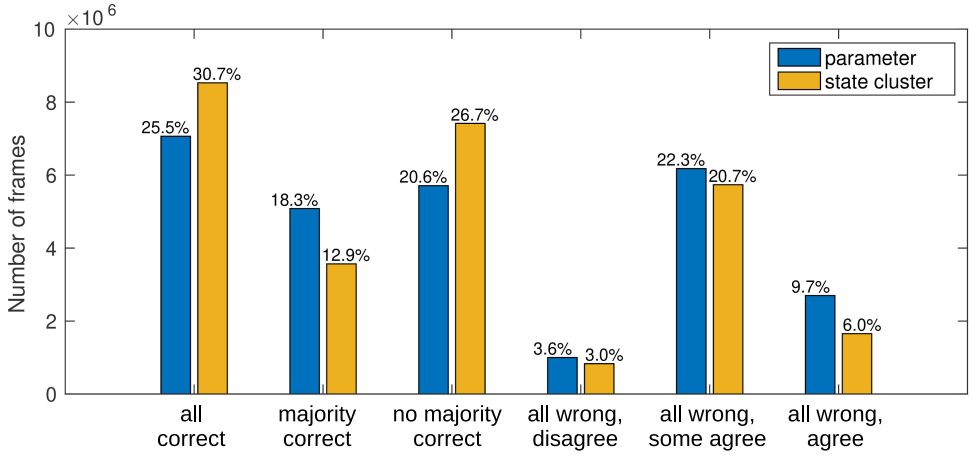


Fig. 9.8 Frame categories of ensembles with model parameter and state cluster diversities, in AMI-IHM. Each ensemble had 4 DNNs, trained toward the $\mathcal{F}_{\text{sMBR}}$ criterion. The ensemble with state cluster diversity has more frames in the *all correct* and *no majority correct* categories.

frames are distributed into the categories, defined in Section 9.2.2, for ensembles with both forms of diversities. When the sets of state clusters differ between systems, the classifications by the systems were considered to be in agreement if there was at least one logical context-dependent state that was common within the state cluster classifications across the systems. The ensemble with state cluster diversity allocates many more frames to the *all correct* and *no majority correct* categories, than the ensemble with model parameter diversity. The category of interest is the *no majority correct* category. The frames within this category are classified correctly by only a minority of the systems. An increase in the number of frames in this category may indicate that the systems with different sets of state clusters

are specialising more specifically toward different frames. The choice of decision tree, that defines how logical context-dependent states are clustered, may affect how easily each frame can be classified.

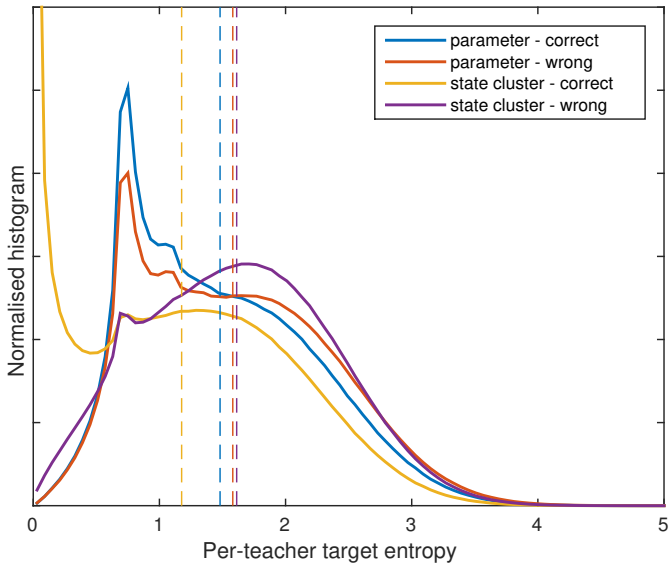


Fig. 9.9 Histograms of per-teacher posterior entropies for frames in the *no majority correct* category, which are classified correctly and wrongly by the teachers in the ensembles with model parameter and state cluster diversities, in AMI-IHM. Dashed lines represent the histogram means. The teachers with different sets of state clusters show different behaviours for correctly and wrongly classified frames in this category.

Figure 9.9 shows histograms of the posterior entropies of each teacher in the ensembles, for frames that have been allocated into the *no majority correct* category. For each ensemble, the entropies of each teacher were divided according to whether that teacher had classified that frame correctly. For the ensemble with model parameter diversity, the distributions of per-teacher entropies are fairly similar, whether the teachers classify the frame correctly or not. However, for the ensemble with state cluster diversity, the entropy distributions for correctly and wrongly classified frames differ greatly. This suggests that by using different sets of state clusters, different systems in the ensemble are each able to better classify different frames. This may suggest that this ensemble relies on its diversity of state cluster sets to obtains a good combined performance.

The distribution of frames shown in Figure 9.8 had been computed when the correctness and agreement of the teachers were measured over the different sets of state clusters that the ensemble had. However, when training a student, the teachers' posteriors are mapped to the student's set of state clusters, using (5.14). The frame categorisation after this mapping is

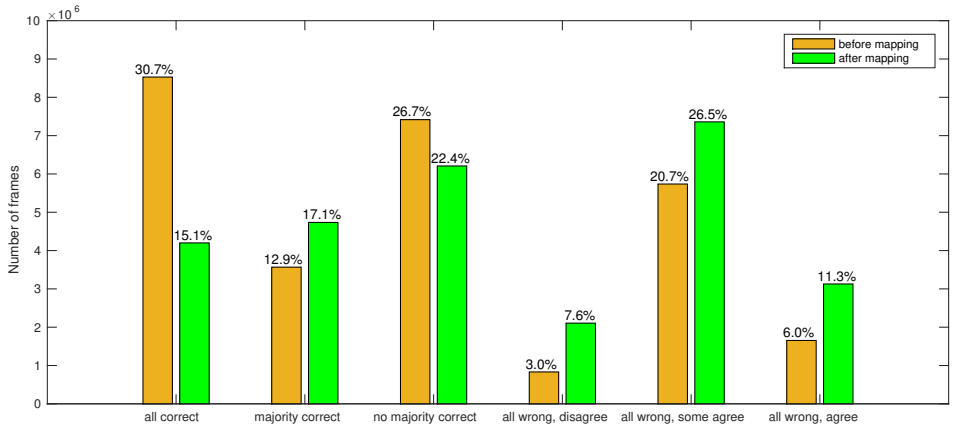


Fig. 9.10 Effect of mapping posteriors across state cluster sets on the frame categories, in AMI-IHM. After mapping, there are fewer frames in the *all correct* and *no majority correct* categories.

shown in Figure 9.10. Here, the correctness and agreement of the targets are measured in the common space of the student's set of state clusters. Although Figure 9.8 shows that there are more frames in the *no majority correct* category in the ensemble with state cluster diversity than the ensemble with parameter diversity before posterior mapping, Figure 9.10 shows that the number of frames in this category decreases after posterior mapping. Mapping the posteriors from multiple sets of state clusters to a single set of state clusters may reduce the ability of the ensemble to express the diversity that it originally had. This may be a limitation caused by the size of the student's set of state clusters.

After posterior mapping, there is also a decrease in the number of frames in the *all correct* category. Before the mapping, a frame was considered to be classified correctly by teacher $\mathbf{\Phi}^m$ if $\arg\max_{s^m} P\left(s^m|\boldsymbol{o}_t, \mathbf{\Phi}^m\right) = \mathcal{T}^m\left(c_t^{\text{ref}}\right)$, where c_t^{ref} is the reference forced alignment logical context-dependent state. As opposed to this, after the posterior mapping, the frame was considered to be classified correctly if $\arg\max_{s^{\mathbf{\Theta}}} \sum_{s^m \in \mathcal{T}^m} P\left(s^{\mathbf{\Theta}}\middle|s^m\right) P\left(s^m|\boldsymbol{o}_t, \mathbf{\Phi}^m\right) = \mathcal{T}^{\mathbf{\Theta}}\left(c_t^{\text{ref}}\right)$. A correct classification under the teacher's set of state clusters does not guarantee correct classification under that of the student. This may be especially evident when the state clusters of $\mathcal{T}^m\left(c_t^{\text{ref}}\right)$ and $\mathcal{T}^{\mathbf{\Theta}}\left(c_t^{\text{ref}}\right)$ only overlap by a few logical context-dependent states. Using a non-approximated form of the posterior mapping of (5.11) may help to alleviate this.

These results point to two issues. First, that the student's set of state clusters may not be able to capture the diverse behaviours of the multiple sets of state clusters used by the teachers. Second, that the approximate posterior mapping may result in incorrect classifications of the targets after mapping. It may be possible to reduce the impact of these issues by using a larger

Table 9.11 Increasing the student's decision tree size, in 207V. The trees with 1000 and 1800 leaves had greedy splits, while that with 15094 leaves was constructed using a Cartesian product of the 4 random forest decision trees of the teachers. Increasing the student's decision tree size brings its performance closer to that of the combined ensemble.

Student decision tree size	No. parameters	Student WER (%) $\mathcal{F}_{\text{TS}}^{\text{CD}\prime}$	Student WER (%) $+\mathcal{F}_{\text{sMBR}}$	Ensemble WER (%)
1000	5.0×10^6	47.3	46.6	
1800	5.8×10^6	47.0	46.3	45.8
15094 (intersect)	19.1×10^6	46.6	46.0	

set of state clusters in the student, and thus increasing its phonetic resolution. The set of state clusters is determined by the leaves of the decision tree. The systems in AMI-IHM used decision trees with 4000 leaves. Increasing the number of leaves further incurs significant additional computational cost during training. As such, this experiment was performed on 207V, where the teachers used decision trees with 1000 leaves. For 207V, a decision tree with 1800 leaves was about the largest that could be generated without having to relax the likelihood thresholds. Using a state cluster set corresponding to the intersect states, formed by the Cartesian product of all of the teachers' decision trees, forms an upper limit to the phonetic resolution that the teacher ensemble can express, as is discussed in Section 3.4.2. Students that used these larger sets of state clusters were trained toward the ensemble with state cluster diversity. The results in Table 9.11 show that using larger sets of state clusters for the student does indeed bring the student's performance closer to that of the combined ensemble. However, using more state clusters leads to a larger student output layer, and thus more parameters. Using a larger student may incur a greater computational cost when performing recognition. Having more parameters may also make it more difficult to train the student to generalise well, but this is not evident in the results. This experiment demonstrates the flexibility allowed by the proposed criterion, in allowing the freedom to choose the student's set of state clusters and output complexity independently of those used by the teachers.

9.6.2 Multi-task teacher-student learning

The results in Table 9.11 suggest that a large decision tree may be required for the student to effectively emulate the ensemble behaviour, when the ensemble has a diversity of state cluster sets. However, this may incur a high computational cost when performing recognition using the student. Rather than compressing the ensemble into a student with a single output layer, the multi-task ensemble topology presented in Section 4.2 can instead be used, where

the hidden layer weights are tied across all members of the ensemble, and each member only has a separate output layer for each different set of state clusters. Since the multi-task ensemble uses all of the different decision trees, it has the same phonetic resolution as an ensemble of separate systems. As opposed to this, a student with a single output layer needs to use intersect states, in order to capture the same phonetic resolution as the ensemble. The intersect states are formed by a Cartesian product of the multiple decision trees. The number of intersect states may be many more than the sum of the leaves of the different decision trees. As such, the multi-task ensemble can capture the same phonetic resolution as an ensemble of separate systems, while potentially having fewer parameters than a student with a single large output layer.

The experiment in Section 8.7 trains a multi-task ensemble using the multi-task cross-entropy criterion of $\mathcal{F}_{\text{MT-CE}}$ in (4.5). It is shown that this yields little diversity between the multi-task output layers, leading to limited gains in combination. The limited diversity in the multi-task ensemble may be because many parameters are shared across the different members of the ensemble. Section 5.1.2 proposes to use teacher-student learning, using the criterion of $\mathcal{F}_{\text{MT-TS}}$ in (5.15), to train a multi-task ensemble to emulate the diverse behaviours of separate systems. This multi-task teacher-student learning method is assessed in this section.

The first experiment compares training a multi-task ensemble using the cross-entropy and teacher-student learning criteria, in the AMI-IHM and 207V datasets. The ensembles used 4 different decision trees, generated using the random forest method. Teacher ensembles of separate DNNs were sequence-trained using these different decision trees. Multi-task ensembles used the same feed-forward hidden layer topologies and multiple decision trees as the ensembles of separate systems. Multi-Task (MT) ensembles were trained using the multi-task cross-entropy criterion, and are referred to as MT. These are the same multi-task ensembles used in Section 8.7. Multi-task ensembles were also trained using the multi-task Teacher-Student (TS) learning criterion, toward the ensembles of separate systems, and are referred to as MT-TS. Further joint sequence discriminative training was performed on the multi-task ensembles using the $\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$ criterion of (4.9). Frame-level combinations of (3.40) and (4.8) were used to combine the ensembles, as it is more computationally efficient than hypothesis-level combination. The joint sequence discriminative training method uses a frame-level combination of the multi-task ensemble, and is therefore matched with the combination method used when performing recognition here. The performances of these ensembles are shown in Table 9.12.

The results suggest that by using teacher-student learning, the combined WER of the multi-task ensemble, of 24.8% in AMI-IHM, can be brought closer to that of the combined

Table 9.12 Multi-task teacher-student learning. All ensembles used 4 random forest decision trees. Combination was performed at the frame level. The combined performance of the multi-task ensemble with teacher-student learning comes close to that of separate systems.

Dataset	Ensemble	Criterion	WER (%) mean single	WER (%) combine	cross-WER (%)
207V	separate	$\mathcal{F}_{\text{sMBR}}$	48.3	46.0	28.4
	MT	$\mathcal{F}_{\text{MT-CE}}$	50.2	49.4	19.6
		$+\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$	48.7	47.8	20.7
	MT-TS	$\mathcal{F}_{\text{MT-TS}}$	47.4	46.3	22.0
		$+\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$	47.1	45.7	22.2
AMI-IHM	separate	$\mathcal{F}_{\text{sMBR}}$	26.0	24.6	15.2
	MT	$\mathcal{F}_{\text{MT-CE}}$	28.6	27.9	11.9
		$+\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$	26.2	25.5	11.7
	MT-TS	$\mathcal{F}_{\text{MT-TS}}$	25.5	24.8	11.5
		$+\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$	25.4	24.4	12.8

ensemble of separate systems of 24.6%, compared to using cross-entropy training yielding a combined WER of 27.9%. This shows that the information propagated over from the separate systems is useful for the multi-task ensemble. Contrary to what was originally hypothesised, learning from diverse separate systems does not yield a consistent increase in the multi-task ensemble diversity. In AMI-IHM, the cross-entropy multi-task ensemble has a cross-WER of 11.9%, which is similar to that trained with teacher-student learning of 11.5%, both of which are less than that for separate systems of 15.2%. This may be a limitation of having many shared parameters across the members of the ensemble. Instead, learning from the separate systems yields an improvement in the performance of each individual member of the multi-task ensemble. The mean single ensemble member performance of the multi-task ensemble with teacher-student learning is 25.5%, which is better than that trained with cross-entropy of 28.6%, in AMI-IHM. This may suggest that propagating information from multiple teachers may help the multi-task ensemble to develop a more general hidden representation. It is this improvement of the single ensemble member performance that seems to be resulting in the better combined performance of the multi-task ensemble.

Performing further sequence discriminative training on the multi-task ensemble after teacher-student learning brings additional performance gains, with the combined multi-task ensembles outperforming the combined ensembles of separate systems in both datasets. This is statistically significant for 207V with a null hypothesis probability less than 0.001, but not for AMI-IHM, with a null hypothesis probability of 0.073. Further sequence discriminative training of the MT-TS ensemble does not result in as much gains as further sequence

discriminative training of the MT ensemble. In AMI-IHM, further sequence discriminative training of the MT-TS ensemble yields a relative gain of 1.6% in the combined performance, while the same for the MT ensemble yields a relative gain of 8.6%. This may indicate that the frame-level posterior information propagated over from the sequence-trained separate systems already conveys some information about their sequence-level behaviours.

Table 9.13 Using a multi-task or single output layer student, in 207V. The ensemble of teachers had 4 DNNs with different random forest decision trees, trained toward the $\mathcal{F}_{\text{sMBR}}$ criterion, and was combined using MBR combination decoding. No sequence training was performed on the students. Using the multi-task topology performs better than a student with a single large output layer, while having fewer parameters.

	No. of parameters	WER (%)
separate ensemble	19.9×10^6	45.8
MT-TS	8.0×10^6	46.3
student with 15094 intersect states	19.1×10^6	46.6
student with 1000 state clusters	5.0×10^6	47.3

In this experiment, the multi-task ensemble is used as a student, to learn from an ensemble of separate systems. This can be contrasted with using a student with a single output layer, as is used in Section 9.6.1. This is shown in Table 9.13 for the 207V dataset. Here, the multi-task ensemble and students with single output layers were trained with only frame-level teacher-student learning, without further sequence discriminative training, to ascertain how well teacher-student learning alone can allow the students to emulate the ensemble performance. The student performances are contrasted against the performance of an ensemble of separate systems, that was combined with hypothesis-level MBR combination decoding of (3.29). This hypothesis-level combination method was used for the ensemble of separate systems, as it represents the best performance that the ensemble can achieve, despite being the most computationally expensive. The results suggest that the multi-task ensemble is able to obtain a performance that is closer to the combined separate systems, while having fewer model parameters than the best student with a single output layer, with a null hypothesis probability less than 0.001. The results suggest that the multi-task ensemble represents a better balance between the number of parameters and the modelling capacity, in terms of the phonetic resolution, than a student with a single large output layer.

The experiments in this section have thus far combined a multi-task ensemble using only frame-level combination. A comparison of various ensemble methods with different forms of combinations is presented in Table 9.14. Here, sequence discriminative training was performed on all of the ensembles. The joint sequence discriminative training method was used for the multi-task ensembles. The HUB4 dataset was also used here for an addi-

tional comparison. Hypothesis-level combination was performed using MBR combination decoding of (3.29). Frame-level combination was performed using (3.40) and (4.8). Another combination method that is compared was to compress each of the ensembles into students with single output layers. These students used greedy decision trees with the same number of leaves as each of the separate systems. Further sequence discriminative training was also performed on these students. The combination methods presented in Table 9.14 are ordered, such that the computational cost of performing recognition decreases when going from the left to the right.

Table 9.14 Multi-task ensemble combination. Further $\mathcal{F}^{\text{joint}}_{\text{MT-sMBR}}$ and $\mathcal{F}_{\text{sMBR}}$ sequence training were performed on the MT-TS ensembles and students respectively. The separate systems can be compressed into a multi-task ensemble using teacher-student learning, without incurring performance degradation.

		Combined WER (%)		
Dataset	Ensemble	hypothesis	frame	student
207V	separate	45.8	46.0	46.6
	MT	47.7	47.8	47.3
	MT-TS	45.7	45.7	46.3
AMI-IHM	separate	24.5	24.6	24.6
	MT	25.4	25.5	25.1
	MT-TS	24.3	24.4	24.6
HUB4	separate	8.7	8.7	9.0
	MT	9.1	9.1	8.8
	MT-TS	8.8	8.7	8.9

The results suggest that the MT-TS ensembles can consistently outperform the MT ensembles. The MT-TS ensembles are able to match, and sometimes even outperform, the ensembles of separate systems. This shows that compressing the separate systems into a multi-task ensemble need not sacrifice the performance, but may in fact lead to improvements. Furthermore, when performing recognition using a multi-task ensemble, data only needs to be fed through the hidden layers once, and should therefore be less computationally expensive than using an ensemble of separate systems. Hypothesis-level combination tends to perform slightly better than frame-level combination in many of the ensembles. However, it is more computationally expensive, as it requires multiple decoding runs. The students of the separate systems and MT-TS ensembles tend to not perform as well as the hypothesis and frame-level combinations, agreeing with the results from the previous experiments.

9.7 Summary

This chapter has presented several experiments to assess the properties of frame-level teacher-student learning. The results suggest that it is the information about the difficulty of classifying each frame, propagated from the teachers, that is useful to the student. Performing further sequence discriminative training brings additional gains to the student, suggesting that the frame-level posteriors do not convey all information about the sequence-level behaviours of sequence-trained teachers.

This chapter has also assessed the ability of students to learn from teachers with different acoustic model topologies and different sets of state clusters. The student learns best from teachers with similar topologies. The proposed method of minimising the KL-divergence between logical context-dependent state posteriors allows a student to learn from an ensemble with a diversity of state cluster sets. However, a large student output layer may be required to effectively capture the phonetic resolution of the ensemble. Teacher-student learning can also be used to train a multi-task ensemble. This allows the multi-task ensemble to learn from diverse separate systems, improving upon using multi-task cross-entropy training. The multi-task ensemble is able to outperform a student with a single large output layer, while having fewer parameters. These results suggest that it is important to carefully design the student to have the capacity to effectively emulate the teachers.

Chapter 10

Experiments on propagating different forms of information

Teacher-student learning can be used to reduce the computational cost of using an ensemble to perform recognition. The experiments in Chapter 9 investigate frame-level teacher-student learning, where information is propagated from the teachers to the student in the form of per-frame state posteriors. The student learns to emulate the ensemble based on this propagated information. There may be other forms of information that are also useful to the student. Section 6.1 has discussed possible methods of propagating information about the hidden layer representations. These methods are assessed in Section 10.1. The student can also be trained to directly emulate the sequence-level behaviours of the teachers, by propagating over sequence-level posterior information, as is discussed in Section 6.2. These methods are investigated in Section 10.2.

10.1 Hidden layer information

In addition to propagating frame-level state cluster posterior information from the teachers to the student, Section 6.1 also discusses the possibility of propagating information about the behaviours of the teachers' hidden layer representations. In particular, this thesis proposes to propagate posterior information about the discriminability of the hidden layer representations, by first training a softmax output layer from the hidden representations of each teacher, then training the student by minimising the KL-divergence between the posteriors at the hidden outputs of the teachers and a hidden output of the student. This section presents a preliminary investigation into the usefulness of this form of information.

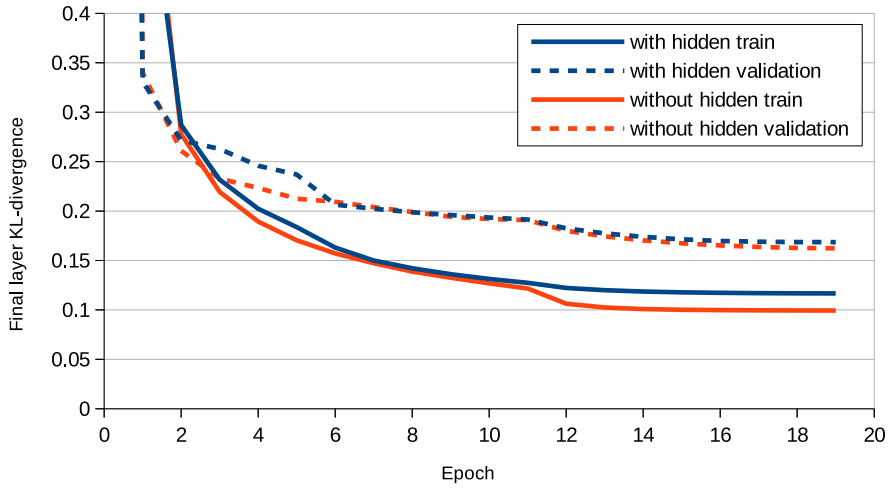
An ensemble of 4 DNN teachers was sequence-trained, in the 207V dataset. These teachers were made different by beginning each training run from a different random parameter initialisation. For each of the hidden layers of each of the teachers, a single layer softmax output was trained using the sequence-level $\mathcal{F}_{\text{sMBR}}$ criterion in (2.83), to provide hidden layer posterior information for the student. A student was trained using both state cluster posterior information at the final output layer and hidden layer posterior information. This was achieved by interpolating together the KL-divergence criterion of $\mathcal{F}^{\text{KL}}_{\text{hid-TS}}$ in (6.4) for each hidden layer, with the standard teacher-student learning criterion of $\mathcal{F}^{\text{state}}_{\text{TS}}$ in (4.15),

$$\mathcal{F}\left(\boldsymbol{\Theta}, \boldsymbol{\Xi}_1, \cdots, \boldsymbol{\Xi}_K\right) = \chi_{K+1}\mathcal{F}^{\text{state}}_{\text{TS}}\left(\boldsymbol{\Theta}\right) + \sum_{k=1}^{K} \chi_k \mathcal{F}^{\text{KL}}_{\text{hid-TS}}\left(\boldsymbol{\Theta}_{1:k}, \boldsymbol{\Xi}_k\right), \tag{10.1}$$

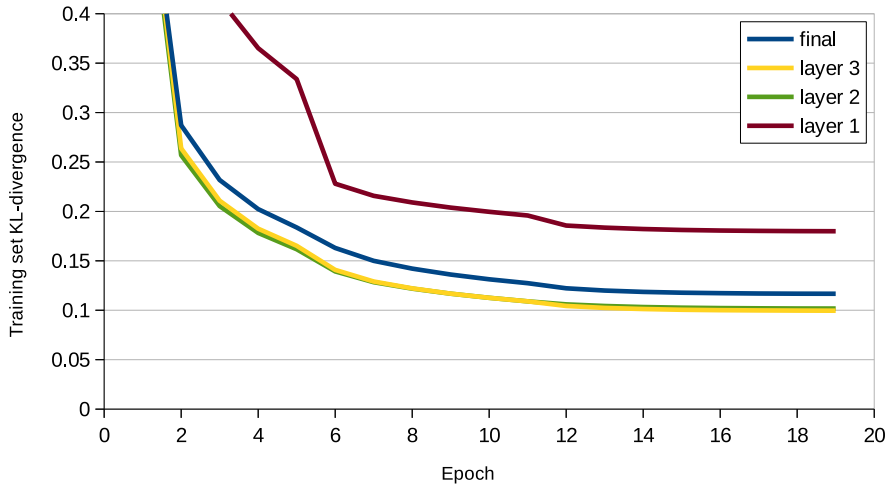
where $\boldsymbol{\Xi}_k$ represents the parameters of the output layer appended to the kth hidden layer in the student, K is the total number of hidden layers, and the interpolation weights satisfy $\chi_k \geq 0$. In this experiment, information was propagated from each of the teachers' multiple hidden layers to each of the corresponding hidden layers in the student. The student was configured to have the same DNN acoustic model topology as each of the teachers, with the same number of layers. Although this simple configuration was used here, it is in general possible to use a student with a different number of layers than the teachers. In this preliminary experiment, equal interpolation weights were used for all layers. However, it may be possible to obtain improved performance by placing more emphasis on later layers, as the student may be better able to emulate the teachers' behaviours after processing data through more layers.

In the frame-level teacher-student learning experiment of Section 9.1 the student NN in the 207V dataset had been pre-trained in a supervised fashion toward the teacher ensemble output layer targets of (4.16). In the current experiment, teacher targets from each hidden layer are available, and can be used to pre-train the student. However, preliminary experiments showed no significant improvements by pre-training each hidden layer of the student toward the corresponding hidden layer targets of the teachers. As such, the same layer-wise pre-training toward the teachers' final output layer targets was used.

Figure 10.1 shows the convergence of the criteria at the various hidden layers, over the epochs of training of the student. Figure 10.1a shows the KL-divergence of the final layer $\mathcal{F}^{\text{state}}_{\text{TS}}$ criterion when training is performed with and without propagating hidden layer posterior information, measured on the training and held-out validation data. Propagating hidden layer posterior information does not appear to improve the ability of the student's final output layer to emulate the teacher ensemble's behaviour. The results in fact suggest a slight degradation in the student performance on the training data when hidden layer posterior information is propagated over. This may suggest that propagating hidden layer posterior



(a) Final layer KL-divergence with and without hidden layer information propagation. Propagating hidden layer information may have a regularisation effect.



(b) Training set KL-divergence for each layer, with hidden layer information propagation. Higher layers are better able to emulate the teachers.

Fig. 10.1 KL-divergence convergence when propagating hidden layer posterior information, in 207V. The student DNN had the same number of layers as each teacher DNN. Hidden layer posterior information was propagated from each teacher hidden layer to the respective student hidden layer.

information may have a regularisation effect on the student. By propagating hidden layer posterior information, the student may be constrained to develop hidden layer representations that follow more closely to those of the teachers.

Figure 10.1b compares the training data KL-divergences for each of the hidden layers, when hidden layer posterior information is being propagated. The results suggest that it

is most difficult for the first hidden layer to emulate the teachers' behaviours. The KL-divergences of the other layers are fairly similar. It may therefore be useful to assign less importance to emulating the first hidden layer, by reducing its criterion interpolation weight.

Table 10.1 Performance of student trained with and without hidden layer posterior information propagation, in 207V. Propagating hidden layer posterior information does not significantly improve the student's performance.

Information	Student WER (%)
final layer	46.9
final layer and hidden layers	46.8

Table 10.1 shows the WER performance of the student trained with and without the propagation of hidden layer posterior information. As a reference, the combined ensemble has a WER of 46.3%. The results suggest that propagating hidden layer posterior information may not significantly benefit the student when used together with frame-level teacher-student learning, will a null hypothesis probability of 0.549. This used equal interpolation weights for all layers. However as mentioned previously, it may be useful to have different interpolation weights for each layer. This may be an interesting direction for future research.

Table 10.2 Sequence training from student initialised with and without hidden layer posterior information propagation, in 207V. Propagating hidden layer posterior information leads to a better initialisation for further sequence training.

Information	Student WER (%) with further $\mathcal{F}_{\text{sMBR}}$ training
final layer	46.6
final layer and hidden layers	46.1

Propagating hidden layer information from an ensemble may allow the student to develop better hidden representations. These better hidden representations may indicate a better initialisation for subsequent training. Table 10.2 shows the performance of further sequence discriminative training of the students, after being initialised through frame-level teacher-student learning, with and without the propagation of hidden layer posterior information. Hidden layer posterior information was not used during the sequence discriminative training phase. Propagating hidden layer posterior information leads to a better student performance after further sequence discriminative training, with a null hypothesis probability less than 0.001. The results suggest that propagating hidden layer posterior information while performing frame-level teacher-student learning may in fact lead to a better initial model for subsequent sequence discriminative training. However, this improved initialisation is not evident in the student after only frame-level teacher-student learning, in Table 10.1.

10.2 Sequence-level information

The standard frame-level teacher-student learning method, discussed in Section 4.3.3, trains the student by propagating per-frame state cluster posterior information from the teachers in the ensemble. It is also possible to propagate information about the hidden layer representations, as is discussed in Section 6.1. However, these forms of information may not effectively convey all of the information about the sequence-level behaviours of the teachers. Furthermore, work in [86] shows that with standard training, sequence-level training can often outperform frame-level training. The experiment in Section 9.3 investigates a simple method of incorporating sequence-level information into the student, by performing further sequence discriminative training on the student, after initial frame-level teacher-student learning. The gains observed when performing further sequence discriminative training on the frame-level student suggest that frame-level teacher-student learning does not effectively convey all information about the sequence-level behaviours of the teachers. Section 6.2 proposes several approaches to directly propagate sequence-level information from the teachers to the student. These sequence-level teacher-student learning methods are investigated in this section.

10.2.1 State cluster sequence posterior

Sequence-level information can be propagated by minimising the KL-divergence between word sequence posteriors. However, the derivative of this criterion can be expensive to compute when training the student, as is discussed in Section 6.2.1. The student can instead be trained by minimising the KL-divergence between state cluster sequence posteriors, using the criterion of $\mathcal{F}^{\text{state}}_{\text{seq-KL}}$ in (6.19). As is discussed in Section 6.2.3, the derivative of this criterion is less expensive to compute, by performing forward-backward operations over lattices whose arcs are marked with state clusters. This section investigates training a student using the proposed $\mathcal{F}^{\text{state}}_{\text{seq-KL}}$ criterion.

An ensemble of 4 sequence-trained DNN teachers was generated by beginning multiple training runs from different random initialisations, in the AMI-IHM dataset. A student, with the same DNN topology as each of the teachers, was trained to emulate the ensemble. The sequence-level teacher-student learning criterion derivative computation in this experiment was implemented using a lattice-based framework. This requires an initial acoustic model to provide acoustic scores to prune the denominator lattices. Frame-level teacher-student learning, with the criterion of $\mathcal{F}^{\text{state}}_{\text{TS}}$ in (4.15), was used to train this initial model, from which sequence-level teacher-student learning was then performed. The denominator lattices of the teachers and student need to have the same support when computing the gradient. In

this experiment, a common support was ensured by using a common set of lattice paths, generated by the initial frame-level student. The teachers' denominator lattices were obtained by rescoring this common lattice with acoustic scores from each of the teachers.

As is discussed in Section 6.2.1, the sequence-level teacher-student learning criteria have a similar KL-divergence form to the standard sequence-level $\mathcal{F}_{\text{MMI}}$ criterion in (6.13). It is therefore simple to interpolate these criteria together when training the student,

$$\mathcal{F} = \chi \mathcal{F}_{\text{MMI}}(\boldsymbol{\Theta}) + (1-\chi)\,\mathcal{F}^{\text{state}}_{\text{seq-TS}}(\boldsymbol{\Theta}), \tag{10.2}$$

where the interpolation weight satisfies $0 \leq \chi \leq 1$. This allows information about the manual transcriptions to be used when training the student.

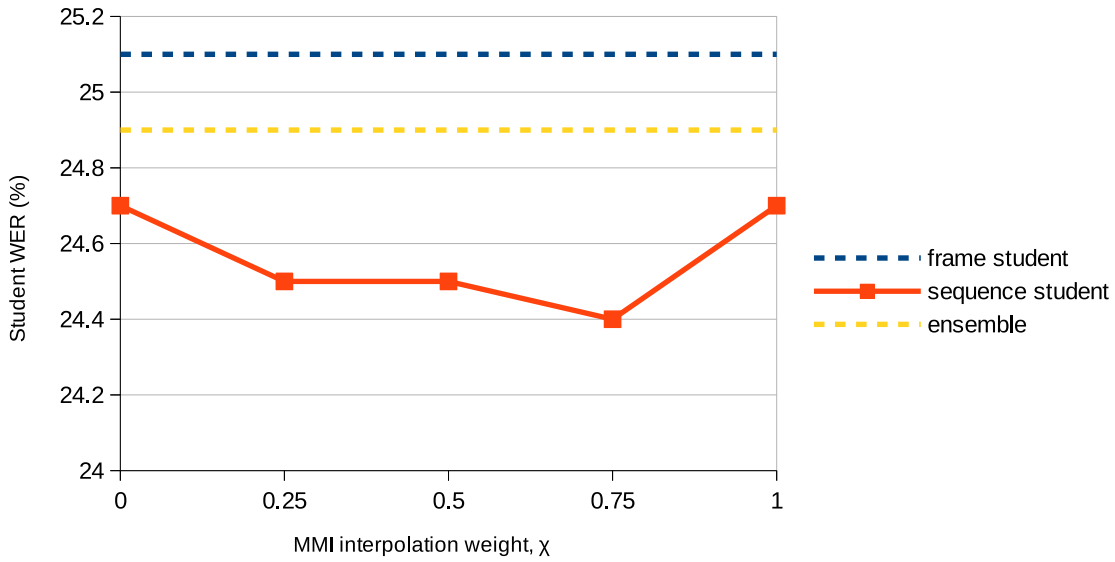


Fig. 10.2 Sequence-level teacher-student learning, in AMI-IHM. A DNN student was trained toward an ensemble of 4 $\mathcal{F}_{\text{sMBR}}$-trained DNN teachers from different random parameter initialisations. The sequence-level student used the frame-level student as the parameter initialisation. The ensemble was combined using MBR combination decoding. The sequence-level student outperforms both the frame-level student and the combined ensemble.

Figure 10.2 shows the performances of students trained with various interpolation weights. The results suggest that using just sequence-level teacher-student learning alone, with $\chi = 0$, outperforms the frame-level student, and performances similarly to further $\mathcal{F}_{\text{MMI}}$ training of the frame-level student. This demonstrates the advantage of propagating sequence-level hypothesis posterior information, over just frame-level state cluster posterior information. Further performance gains are obtained by interpolating together the sequence-level teacher-student learning and $\mathcal{F}_{\text{MMI}}$ criteria, suggesting that including information about the manual

transcriptions can be beneficial to the student. This differs from the trend observed when interpolating the cross-entropy and frame-level teacher-student learning criteria in Section 9.1, where cross-entropy interpolation is found not to help when the teachers have been sequence-trained. The sequence-level students here are able to outperform the combined teacher ensemble.

Table 10.3 Comparing sequence-level teacher-student learning with further $\mathcal{F}_{\text{sMBR}}$ training of a frame-level student, in AMI-IHM. The sequence-level student can outperform further $\mathcal{F}_{\text{sMBR}}$ training on the frame-level student.

Training	WER (%)
$\mathcal{F}_{\text{TS}}^{\text{state}}$	25.1
$\mathcal{F}_{\text{TS}}^{\text{state}} + \mathcal{F}_{\text{sMBR}}$	24.8
$\mathcal{F}_{\text{TS}}^{\text{state}} + \mathcal{F}_{\text{seq-TS}}^{\text{state}}$	24.4
Combined ensemble	24.9

The sequence-level student is compared to performing further sequence-level $\mathcal{F}_{\text{sMBR}}$ training on the frame-level student, as is done in Section 9.3. Interpolation of the $\mathcal{F}_{\text{MMI}}$ criterion with a weight of $\chi = 0.75$ was used when performing sequence-level teacher-student learning. The results in Table 10.3 show that the sequence-level student outperforms both further $\mathcal{F}_{\text{sMBR}}$ training of the frame-level student and the combined teacher ensemble, with null hypothesis probabilities of less than 0.001.

10.2.2 State cluster diversity

The experiments in Section 10.2.1 show that sequence-level information can be propagated to perform teacher-student learning, by minimising the KL-divergence between state cluster sequence posteriors. As is described in Section 6.2.3, the derivative of this criterion can be computed efficiently over lattices whose arcs are marked with state clusters. However, the KL-divergence requires the same support for all distributions, and therefore all systems are restricted to use the same set of state clusters. This limits the forms of diversities that are allowed in a teacher ensemble. It is possible to allow additional forms of diversities by instead marking the arcs with words or sub-word units, and considering KL-divergences over their respective sequence posteriors. However, these choices of arc markings require two levels of forward-backward operations to compute the criteria derivatives, as opposed to the single level required when marking the arcs with state clusters. Section 6.2.4 instead proposes to mark the arcs with logical context-dependent states, and propagate sequence-level posterior information by minimising the KL-divergence between logical context-dependent state sequence posteriors, using the criterion of $\mathcal{F}_{\text{seq-TS}}^{\text{CD}}$ in (6.26). This allows the teacher

ensemble to use a diversity of state cluster sets, while still allowing the criterion derivative to be simply and efficiently computed using a single level of forward-backward operations. The current section investigates the effectiveness of using the proposed $\mathcal{F}^{\text{CD}}_{\text{seq-TS}}$ criterion to train a student to emulate an ensemble with a diversity of state cluster sets.

When performing teacher-student learning at the frame level, the teacher ensemble can be allowed to have a diversity of state cluster sets by using the criterion of $\mathcal{F}^{\text{CD}\prime}_{\text{TS}}$ in (5.5). As is described in Section 5.1.1, this criterion is motivated by minimising the KL-divergence between per-frame logical context-dependent state posteriors. The experiments in Section 9.6.1 show that this criterion can train a student to emulate an ensemble with a diversity of state cluster sets. However, the information propagated by frame-level training may not adequately capture the sequential nature of the data. Furthermore, this frame-level training method requires the approximation of (5.11) to compute the targets when training the student. The sequence-level criterion propagates information about the sequence-level behaviours of the teachers, and also does not require any approximations when computing the targets to train the student. This experiment compares frame and sequence-level teacher-student learning toward an ensemble with a diversity of state cluster sets. A lattice-based implementation of the sequence-level criterion derivative computation was used here, which requires an initial acoustic model to provide acoustic scores to prune the denominator lattices. The frame-level student was again used as the initialisation for sequence-level teacher-student learning. The denominator lattice arcs were marked with intersect states, formed by a Cartesian product of all of the decision trees of the teachers and student.

An ensemble of 4 sequence-trained DNNs was generated, each having a different decision tree, obtained using the random forest method, in AMI-IHM. Students, with the same DNN topology as each teacher, were trained toward this ensemble using frame and sequence-level teacher-student learning. The students used a decision tree of the same size as those of the teachers, but that was trained with greedy splits. Unlike in Section 10.2.1, no $\mathcal{F}_{\text{MMI}}$ interpolation was used here, to ascertain the sole contribution from sequence-level teacher-student learning.

Table 10.4 compares the performances of these frame and sequence-level students. The results show that by using sequence-level teacher-student learning, the student is able to more closely approach the combined performance of the ensemble, than can a frame-level student. The sequence-level student significantly outperforms the frame-level student, with a null hypothesis probability less than 0.001.

The difference between the combined ensemble performance and that of the sequence-level student is not significant, with a null hypothesis probability of 0.412. In the experiment in Section 9.6.1, frame-level teacher-student learning is shown to result in a degraded student

Table 10.4 Comparing frame and sequence-level teacher-student learning with different sets of state clusters, in AMI-IHM. The ensemble had 4 $\mathcal{F}_{\text{sMBR}}$-trained DNNs with different random forest decision trees. The DNN student used the greedy decision tree. The sequence-level student used the frame-level student as the parameter initialisation. Sequence-level teacher-student learning brings the student performance closer to that of the combined ensemble.

System	WER (%)
frame-level student, $\mathcal{F}_{\text{TS}}^{\text{CD}\prime}$	25.5
sequence-level student, $\mathcal{F}_{\text{seq-TS}}^{\text{CD}}$	24.6
Combined ensemble	24.5

performance. It is hypothesised that the degradation either arises from the approximation used to map the posterior targets between different sets of state clusters or from the limited phonetic resolution of the student. In sequence-level teacher-student learning, no approximations are needed when computing the target posteriors. The results in Table 10.4 suggest that despite its limited phonetic resolution, the sequence-level student is able to effectively emulate the combined ensemble behaviour. The student here used a greedy decision tree with 4000 leaves, identical in size to each of the teachers' decision trees. However, a performance degradation due to the limited phonetic resolution of the sequence-level student is observed in the lattice-free experiments in Section 11.6.2.

10.3 Summary

This chapter has investigated propagating information in the forms of hidden layer posteriors and sequence posteriors. The preliminary experiments suggest that propagating hidden layer posterior information can allow the student to develop better hidden representations, which are beneficial when performing further sequence discriminative training. Sequence-level information is shown to allow the student to better emulate the teacher ensemble. The sequence-level criteria of the KL-divergences between either state cluster sequence or logical context-dependent state sequence posteriors have been assessed. The latter of which can be used when the ensemble has a diversity of state cluster sets.

Chapter 11

Experiments on extensions for lattice-free systems

In the previous experiments, sequence discriminative training and sequence-level teacher-student learning are implemented using a lattice-based approach to compute the derivatives of the criteria. Here, the lattices representing the competing hypotheses allowed by the recognition graph are pruned to reduce the computational cost. The lattice-free method, described in Section 2.5.2, is an alternative method to reduce the computational cost. Here, the lattices are not pruned. Instead, the number of lattice arcs are reduced by simplifying the recognition graph to use a phone-level language model, rather than a word-level language model. Other simplifications that are often made to further reduce the computational cost are to use a simpler 2-state HMM topology and a slower 30ms frame shift. These unpruned lattices are also generated on-the-fly from the recognition graph during training. Work in [116] shows that sequence discriminative training using this lattice-free method can perform competitively with lattice-based training.

This chapter extends the ensemble generation and teacher-student learning methods to use lattice-free systems. Sections 11.2, 11.3 and 11.4 show transitions of the single system and ensemble performances from lattice-based to lattice-free training. Section 11.5 assess methods of performing frame-level teacher-student learning when using the lattice-free model topology. Sections 11.6 investigates a lattice-free implementation of the proposed sequence-level teacher-student learning methods. Finally in Section 11.7, a rich ensemble is generated with multiple forms of diversities, and various schemes of compressing this ensemble are assessed.

11.1 Setup

The lattice-free experiments were performed on the AMI-IHM and MGB-3 datasets. A 4-gram phone-level language model was used for sequence discriminative training. A left-biphone context-dependence and 2-state HMM topology shown in Figure 2.2b were used, with decision trees having 2000 and 3600 leaves for AMI-IHM and MGB-3 respectively. A 30ms shift was used between consecutive frames. Initial experiments found that using an acoustic model that captures a longer temporal context span leads to better performance. The acoustic models here used interleaved layers of TDNNs and LSTMs, referred to as the TDNN-LSTM topology. For both datasets, the TDNN layers consisted of 600 nodes with ReLU activations. Each TDNN layer spliced together the activations of the preceding layer over a finite context of past and future frames. The splice windows sub-sampled the frames from the preceding layer to limit the number of parameters, similarly to the TDNN topology in [112]. The LSTM layers consisted of 512 cells, with 128 recurrent and 128 non-recurrent projections. From the input to the output, the layers were arranged with splice windows described as $\{-2,-1,0,1,2\}$ $\{-1,0,1\}$ L $\{-3,0,3\}$ $\{-3,0,3\}$ L $\{-3,0,3\}$ $\{-3,0,3\}$ L, where L indicates an LSTM layer. The TDNN-LSTM model parameters were not pre-trained, but began directly from random initialisation. Since no initial acoustic scores are required for lattice pruning in the lattice-free method, there is no need for initial frame-level cross-entropy training. As such, the systems were trained directly toward the sequence-level $\mathcal{F}_{\text{MMI}}$ criterion of (2.81). Furthermore, since state cluster posteriors are no longer required without the need for cross-entropy training, the lattice-free acoustic models were configured with linear output layers that directly produce log-acoustic scores, such that the hypothesis posteriors can be computed using (2.120). An exponentially decaying learning rate schedule was used, with the decay rate and number of epochs determined by initial hyper-parameter sweeps. The work in [116] suggests that regularisation is important for lattice-free training. Regularisation was implemented to minimise the L2 norm of the NN linear output activations, and also to minimise a measure of the saturation of the NN nonlinearities. A secondary softmax output layer was also trained by minimising the KL-divergence between these secondary state cluster posteriors and the soft alignments from the numerator lattice. This provides an additional form of supervision, to aid in learning. The additional output layer was not used when performing recognition. The denominator lattices were augmented with the leaky HMM topology, described in [116]. The experimental configurations used for the systems in this chapter differ significantly from those in the previous chapters. As such, the experimental results in this chapter may not be comparable to those in the previous chapters.

11.2 Single system performance

The first experiment compares the performances of single systems trained with either lattice-based or lattice-free implementations of sequence discriminative criteria. The TDNN-LSTM acoustic model topology was used for all systems. The lattice-based systems used a 3-state HMM topology in Figure 2.2a, a triphone context, a frame shift of 10ms, and decision trees with 4000 and 9200 leaves for AMI-IHM and MGB-3 respectively. These were first trained using the frame level cross-entropy criterion of $\mathcal{F}_{\text{CE}}$ in (2.80), then sequence discriminative training was performed using the lattice-based method toward either the $\mathcal{F}_{\text{sMBR}}$ criterion of (2.83) or the $\mathcal{F}_{\text{MMI}}$ criterion of (2.81). Lattice-free systems were trained beginning from random parameter initialisations, as is described in Section 11.1.

Table 11.1 Comparing lattice-based and lattice-free single systems. All systems used TDNN-LSTM acoustic models.

	Lattice-based WER (%)			Lattice-free WER (%)
Dataset	$\mathcal{F}_{\text{CE}}$	$\mathcal{F}_{\text{sMBR}}$	$\mathcal{F}_{\text{MMI}}$	$\mathcal{F}_{\text{MMI}}$
AMI-IHM	26.7	24.8	24.9	25.3
MGB-3	26.0	23.9	23.7	23.6

Table 11.1 compares these lattice-based and lattice-free systems. The results show that lattice-free training is able to yield a performance that is competitive with lattice-based training in the MGB-3 dataset. However, in AMI-IHM, the lattice-based sequence-level systems outperform the lattice-free system, with a null hypothesis probability less than 0.001 for the $\mathcal{F}_{\text{sMBR}}$ system. This performance degradation arises because of the use of the simpler context-dependence and HMM topology, and smaller decision tree, in the lattice-free system. A lattice-free system that used triphones, with a 3-state HMM, and 4000 decision tree leaves in AMI-IHM has WER of 24.8%. Despite this performance degradation, the simpler topologies were used for the lattice-free systems in this thesis, to reduce the computational cost in both training and recognition.

11.3 Ensemble diversity

Ensembles of lattice-free systems may have the advantage of greater diversity. When generating an ensemble of lattice-based systems, each system is first trained with the cross-entropy criterion, before sequence discriminative training. The targets for cross-entropy training are often obtained from a common source, which in this thesis is a GMM-HMM system. As such, all systems in the ensemble may exhibit a common bias toward the

GMM-HMM system behaviours. Furthermore, the pruned lattices in lattice-based sequence discriminative training only consider a limited variety of possible transition times, which may limit how far the time alignment behaviour of the sequence-trained system can diverge from that of the cross-entropy system. These may limit the ensemble diversity. By not performing initial cross-entropy training, lattice-free systems are not biased in this manner, as is discussed in Section 2.5.2. Furthermore, the unpruned lattices contain a full diversity of all of the hypotheses that are allowed by the lattice-free recognition graph. This may allow the lattice-free systems to develop more diverse behaviours.

To demonstrate the biasing of the systems toward the cross-entropy targets, ensembles of 4 systems were trained, first with the cross-entropy criterion, then with a lattice-based implementation of sequence discriminative training, in the AMI-IHM dataset. The systems were made different by beginning multiple training runs from different random parameter initialisations. The ensembles consisted of either all DNN or all BLSTM acoustic models, with the topologies described in Table 7.2. The greater BLSTM diversity measured in Section 8.1.3 suggests that BLSTMs are more flexible than DNNs. They may therefore be more likely to specialise toward the forced alignment behaviours after cross-entropy training. Table 11.2 measures the frame and hypothesis-level single system performances and diversities, on the *eval* data. The frame-level performance was measured using the Frame Error Rate (FER), with reference forced alignments of the *eval* data obtained using the same GMM-HMM system as was used to obtain forced alignments of the training data. The frame-level diversity was measured using the cross-FER in (3.65).

Table 11.2 Bias of lattice-based systems toward cross-entropy forced alignments, in AMI-IHM. For each topology type, 4 acoustic models were trained, beginning from different random parameter initialisations. The BLSTMs have smaller mean FERs and cross-FERs, suggesting that they behave more similarly to the forced alignments.

Ensemble	Training	mean FER (%)	cross-FER (%)	mean WER (%)	cross-WER (%)
DNN	$\mathcal{F}_{\text{CE}}$	51.8	35.9	28.4	11.4
	$+\mathcal{F}_{\text{sMBR}}$	63.8	47.8	25.7	11.8
BLSTM	$\mathcal{F}_{\text{CE}}$	40.9	29.2	25.8	17.6
	$+\mathcal{F}_{\text{sMBR}}$	46.9	35.6	25.1	19.4

The results show that as expected, the more flexible BLSTMs are able to behave more similarly to the GMM-HMM forced alignments after cross-entropy training, by having a lower mean FER of 40.9%, compared to that of the DNNs of 51.8%. This suggests that the BLSTM systems may be specialising more toward the behaviour of the GMM-HMM system. At the same time, the cross-FER diversity between the BLSTMs of 29.2% is much lower than that between the DNNs of 35.9%, indicating a stronger common bias toward the

GMM-HMM alignments. With sequence discriminative training, the cross-FERs for the DNNs and BLSTMs increase from 35.9% to 47.8% and 29.2% to 25.6% respectively. These larger cross-FER diversities after performing sequence discriminative training suggest that the systems diverge slightly away from their common bias. However, even after sequence discriminative training, the cross-FER between the BLSTMs of 35.6% is still much smaller than that between the DNNs of 47.8%. As such, sequence discriminative training may not be able to fully overcome the common bias. These results suggest that the impact of a common bias toward the cross-entropy forced alignments is more evident when the acoustic models are more flexible. However, the experiment in Section 8.1.3 suggests that more flexible topologies may be preferred when generating an ensemble, to obtain a wider diversity of behaviours. The lattice-free method allows these more flexible acoustic model topologies to be used, without incurring additional bias toward cross-entropy forced alignments.

The lower cross-FER diversity between the BLSTMs is not reflected in the hypothesis-level cross-WER diversity. This may suggest that the long span temporal modelling of the BLSTM is able to more dynamically interact with the alignment and language models, to produce a wider variety of 1-best hypotheses.

The next experiment compares the diversity and combination gains that can be obtained from ensembles of either lattice-free or lattice-based systems. Ensembles of 4 TDNN-LSTM systems were generated, each beginning from a different random parameter initialisation. All system in each ensemble were trained toward either a lattice-free implementation of the $\mathcal{F}_{\text{MMI}}$ criterion, or were first trained using the cross-entropy criterion, followed by sequence discriminative training using a lattice-based implementation of the $\mathcal{F}_{\text{sMBR}}$ criterion. The performances of these ensembles are shown in Table 11.3, for the AMI-IHM and MGB-3 datasets. The ensembles were combined using hypothesis-level MBR combination decoding of (3.29). Although the comparison here is made between lattice-based $\mathcal{F}_{\text{sMBR}}$ and lattice-free $\mathcal{F}_{\text{MMI}}$ ensembles, which use different training criteria, the results in Table 11.1 indicate that, at least for single systems, lattice-based $\mathcal{F}_{\text{MMI}}$ and $\mathcal{F}_{\text{sMBR}}$ systems perform similarly.

The results show that the lattice-free ensembles do indeed exhibit greater diversity than the cross-entropy or lattice-based sequence-trained ensembles. In AMI-IHM, the lattice-free ensemble has a cross-WER of 18.1%, which is larger than those of the cross-entropy and lattice-based sequence-trained ensembles of 16.4% and 15.9% respectively. This leads to a larger relative combination gain of 12.3% for the lattice-free ensemble, compared with 9.3% for the lattice-based sequence-trained ensemble in AMI-IHM. The observed greater diversity within the lattice-free ensemble may be a result of the lack of initial cross-entropy training or because of the greater diversity of hypotheses captured within the denominator lattices. However, many differences exist between the lattice-based and lattice-free systems, and it is

Table 11.3 Random initialisation ensembles trained using lattice-based and lattice-free methods. All systems used the TDNN-LSTM topology. Combination was performed using MBR combination decoding. Lattice-free ensembles exhibit greater diversity and combination gains.

Dataset	Training	Single system WER (%) mean	best	worst	std dev	Combined WER (%)	cross-WER (%)
AMI-IHM	$\mathcal{F}_{\text{CE}}$	26.7	26.6	26.9	0.15	24.4	18.3
	lattice-based $\mathcal{F}_{\text{sMBR}}$	24.8	24.6	25.0	0.17	22.5	17.2
	lattice-free $\mathcal{F}_{\text{MMI}}$	25.3	25.1	25.4	0.15	22.2	20.8
MGB-3	$\mathcal{F}_{\text{CE}}$	26.0	25.8	26.2	0.17	24.2	16.4
	lattice-based $\mathcal{F}_{\text{sMBR}}$	23.9	23.8	24.0	0.10	21.9	15.9
	lattice-free $\mathcal{F}_{\text{MMI}}$	23.6	23.5	23.7	0.10	20.8	18.1

difficult to determine the main sources of diversity contributions without further analysis. This may be an interesting direction for future research.

Acoustic model parameter diversity is incorporated into the ensembles in Table 11.3, by beginning multiple training runs from different random parameter initialisations. It may also be possible to leverage upon the diversity between the behaviours of lattice-based and lattice-free systems. As has been previously discussed, lattice-based systems may exhibit a strong bias toward the initial cross-entropy forced alignments, while lattice-free systems do not. This may suggest that lattice-based and lattice-free systems behave differently from each other. Furthermore, the lattice-based and lattice-free systems used in this thesis employ different context-dependencies, HMM topologies, decision tree sizes, and frame shifts. These may all contribute to differences between their behaviours.

Table 11.4 Combination of lattice-based and lattice-free TDNN-LSTM systems. Combination was performed using MBR combination decoding, with additional zero-weight states begin interpolated to correct for the different frame rates. Lattice-based and lattice-free systems have highly diverse behaviours.

Dataset	Single system WER (%) lattice-based $\mathcal{F}_{\text{sMBR}}$	lattice-free $\mathcal{F}_{\text{MMI}}$	Combined WER (%)	cross-WER (%)
AMI-IHM	24.8	25.3	22.5	36.8
MGB-3	23.9	23.6	21.6	20.2

Table 11.4 shows the performance of combining one lattice-based system with one lattice-free system. The hypothesis-level MBR combination decoding method was used, and additional states with zero weights were interpolated into the decoding lattices of the lattice-free systems, to correct for the difference in frame rate from the lattice-based systems. The results suggest that the lattice-based and lattice-free systems are highly diverse in their

behaviours, with large cross-WERs between them. This is accompanied by large WER performance gains in combination. This experiment therefore demonstrates how multiple forms of diversities can be incorporated into an ensemble, simply by using different training frameworks.

11.4 Ensemble combination methods

The lattice-free ensembles in the previous experiments were combined using hypothesis-level MBR combination decoding. However, this can be computationally expensive, as multiple decoding runs need to be performed. It is less computationally expensive to perform combination at the frame-level, as this requires only a single decoding run for the whole ensemble. The frame-level combination methods discussed in Section 3.4.2 combine over either state cluster posteriors or scaled observation likelihoods. As is described in Section 2.5.2, it is common when using lattice-free training to have an NN acoustic model topology with linear outputs that can be interpreted as log-acoustic scores. As such, state cluster posteriors are not directly obtainable for frame-level combination. In a standard hybrid NN-HMM system, the scaled observation likelihoods, which can be combined over, are interpreted as acoustic scores. Therefore, in a lattice-free ensemble, frame-level combination can also be performed over the acoustic scores. As is discussed in Section 3.4.2, there are many possible methods of combination, such as taking a sum or product between the systems. A frame-level combination of a product over acoustic scores can be expressed as

$$\mathcal{A}\left(\mathbf{o}_t, s, \widehat{\mathbf{\Phi}}\right) = \prod_{m=1}^{M} \mathcal{A}^{\lambda_m}\left(\mathbf{o}_t, s, \mathbf{\Phi}^m\right), \tag{11.1}$$

where $\mathcal{A}\left(\mathbf{o}_t, s, \mathbf{\Phi}^m\right)$ are the acoustic scores of the mth system, obtained by taking the exponential of the lattice-free NN acoustic model outputs. Here, the interpolation weights satisfy $\lambda_m \geq 0$.

A sum combination over acoustic scores for lattice-free systems is not as trivial. The linear outputs of the NN, which represent log-acoustic scores, are unbounded. Therefore taking their exponential to form the acoustic scores can have vastly different dynamic ranges between the systems. One possible approach to ensure that the scores being combined operate over similar dynamic ranges is to normalise the acoustic scores from each of the systems before they are combined,

$$\mathcal{A}\left(\mathbf{o}_t, s, \widehat{\mathbf{\Phi}}\right) = \sum_{m=1}^{M} \lambda_m \frac{\mathcal{A}\left(\mathbf{o}_t, s, \mathbf{\Phi}^m\right)}{\sum\limits_{s' \in \mathcal{T}} \mathcal{A}\left(\mathbf{o}_t, s', \mathbf{\Phi}^m\right)}, \tag{11.2}$$

where the interpolation weights satisfy $\lambda_m \geq 0$. This per-system normalisation may help to prevent a single system from dominating the sum combination, and is similar to the method used in (5.18), to obtain the targets for frame-level teacher-student learning using lattice-free systems.

Table 11.5 Lattice-free ensemble combination methods. The Ensembles had 4 TDNN-LSTMs, beginning from different random parameter initialisations. Equal interpolation weights were used.

	Combined WER (%)	
Combination method	AHI-IHM	MGB-3
mean single	25.3	23.6
hypothesis	22.2	20.8
acoustic score sum	21.9	20.9
acoustic score product	21.8	21.0

Table 11.5 compares the different methods of combining lattice-free ensembles, using the same ensembles from Section 11.3. Surprisingly, the results in AMI-IHM show different trends than when combining ensembles of lattice-based systems in Table 8.10. In AMI-IHM, the frame-level combination methods perform significantly better than hypothesis-level combination, with null hypothesis probabilities less than 0.001. However, this trend is not emulated in the MGB-3 dataset.

11.5 Frame-level teacher-student learning

Although frame-level combination is less computationally expensive than hypothesis-level combination when performing recognition, it still requires data to be fed through each of the separate acoustic models. Teacher-student learning can be used to further reduce this computational cost, by training a single student to emulate the combined ensemble. Standard frame-level teacher-student learning, discussed is Section 4.3.3, trains the student by minimising the KL-divergence between state cluster posteriors. However, as is discussed in Section 2.5.2, lattice-free systems often use NN acoustic models with linear outputs that are interpreted as log-acoustic scores. State cluster posteriors are therefore not directly obtainable in these systems.

Section 5.2 describes several possible extensions to frame-level teacher-student learning to allow lattice-free systems to be used. One possible method of training the student is to minimise the Mean Squared Error (MSE) between the log-acoustic scores [79], using the criterion of $\mathcal{F}^{\text{MSE}}_{\text{LF-TS}}$ in (5.17). Another method, proposed in this thesis, is to first normalise the

acoustic scores of each system, then minimise the KL-divergence between these normalised scores, using the criterion of $\widetilde{\mathcal{F}}^{\text{KL}}_{\text{LF-TS}}$ in (5.18). This KL-divergence criterion allows the existing frame-level teacher-student learning infrastructure to be used, because of its similarity to the standard criterion. These frame-level teacher-student learning methods are compared in this section.

Table 11.6 Frame-level teacher-student learning with lattice-free systems, in AMI-IHM. TDNN-LSTM students were trained toward an ensemble of 4 TDNN-LSTM teachers from different random parameter initialisations. The ensemble was combined using MBR combination decoding. The proposed frame-level criterion of $\widetilde{\mathcal{F}}^{\text{KL}}_{\text{LF-TS}}$ allows the student to learn better from the ensemble than using $\mathcal{F}^{\text{MSE}}_{\text{LF-TS}}$.

Training	WER (%)
$\mathcal{F}_{\text{MMI}}$	25.3
$\mathcal{F}^{\text{MSE}}_{\text{LF-TS}}$	27.0
$\widetilde{\mathcal{F}}^{\text{KL}}_{\text{LF-TS}}$	22.9
Combined ensemble	22.2

An ensemble of 4 lattice-free TDNN-LSTMs was generated, by beginning multiple training runs from different random parameter initialisations, in AMI-IHM. Students with the same TDNN-LSTM topology as each teacher were trained using both variants of the frame-level teacher-student learning criteria, toward this ensemble. Equal interpolation weights were used. The performances of these students are shown in Table 11.6. The ensemble was combined using hypothesis-level MBR combination decoding of (3.29).

The results suggest that using the MSE-based criterion may not effectively propagate information from the teachers to the student. On the other hand, the proposed KL-divergence-based criterion is able to bring the student performance closer to that of the combined ensemble, outperforming a single lattice-free $\mathcal{F}_{\text{MMI}}$ system.

11.6 Sequence-level teacher-student learning

Frame-level teacher-student learning may not effectively propagate all information about the sequence-level behaviours of the teachers. The experiments in Section 10.2 show that sequence-level teacher-student learning can propagate over information that is more useful to the student. Section 10.2 uses a lattice-based implementation of sequence-level teacher-student learning. The same sequence-level teacher-student learning criteria can be implemented within a lattice-free framework. This section assesses lattice-free implementations of sequence-level teacher-student learning.

11.6.1 State cluster sequence posterior

Sequence-level posterior information can be propagated by using the criterion of $\mathcal{F}^{\text{state}}_{\text{seq-TS}}$ in (6.19). This minimises a KL-divergence between state cluster sequence posteriors. When the derivative of this criterion is computed using the lattice-free method, no initial acoustic model is required to produce acoustic scores for lattice pruning. Therefore, training of the student can begin from a random parameter initialisation. As opposed to this, the lattice-based method explored in Section 10.2 used the frame-level student as the initial acoustic model. This current section compares performing lattice-free sequence-level teacher-student learning, with and without initialising the student using frame-level teacher-student learning.

The lattice-free ensembles of 4 TDNN-LSTMs, generated from multiple random parameter initialisations, were used as the teacher ensembles, in the AMI-IHM and MGB-3 datasets. Students with the same TDNN-LSTM acoustic model topology as each teacher were trained toward these ensembles. As with lattice-free $\mathcal{F}_{\text{MMI}}$ training, the learning rates and number of epochs used to train the student were chosen using initial hyper-parameter sweeps. Initial tests found that regularisation by minimising the KL-divergence between the secondary softmax output layer posteriors and the numerator lattice soft alignments did not aid the student. Initial experiments also found that unlike the lattice-based implementation in Section 10.2.1, $\mathcal{F}_{\text{MMI}}$ interpolation yielded no additional gains in the lattice-free implementation. These were therefore not used when training the student. The denominator lattices from both the student and teachers were augmented with the leaky HMM topology, as this was found to improve the student performance. The performance of these students is compared in Table 11.7. As a reference, the combined ensembles have WERs of 22.2% for AMI-IHM and 20.8% for MGB-3.

The results show that by using a lattice-free implementation of sequence-level teacher-student learning, the student can be trained such that its performance is closer to that of the teacher ensemble than both a frame-level student and a single system trained with lattice-free $\mathcal{F}_{\text{MMI}}$. The lattice-free framework allows the student to be trained, beginning from a random parameter initialisation. The results in AMI-IHM may suggest that initialising the student using frame-level teacher-student learning yields a slightly better sequence-level student performance than starting from a random parameter initialisation. However, this may not be statistically significant, with a null hypothesis probability of 0.276. The trend in MGB-3 is the opposite, where starting from a random parameter initialisation yields slightly better sequence-level student performance. However, this again may not be statistically significant, with a null hypothesis probability of 0.097. Initialising the student's parameters with frame-level teacher-student learning may be more useful in AMI-IHM, which has less training data than MGB-3.

Table 11.7 Lattice-free sequence-level teacher-student learning. TDNN-LSTM students were trained toward ensembles of 4 TDNN-LSTM teachers from different random parameter initialisations. Training of the sequence-level students began either from the frame-level students or random parameter initialisations. Sequence-level teacher-student learning is able to bring the student performance closer to that of the combined ensemble, than can frame-level teacher-student learning.

Dataset	Training	WER (%)
AMI-IHM	$\mathcal{F}_{\text{MMI}}$	25.3
	$\widetilde{\mathcal{F}}^{\text{KL}}_{\text{LF-TS}}$	22.8
	$\widetilde{\mathcal{F}}^{\text{KL}}_{\text{LF-TS}} + \mathcal{F}^{\text{state}}_{\text{seq-TS}}$	22.5
	rand init $+ \mathcal{F}^{\text{state}}_{\text{seq-TS}}$	22.7
MGB-3	$\mathcal{F}_{\text{MMI}}$	23.6
	$\widetilde{\mathcal{F}}^{\text{KL}}_{\text{LF-TS}}$	23.0
	$\widetilde{\mathcal{F}}^{\text{KL}}_{\text{LF-TS}} + \mathcal{F}^{\text{state}}_{\text{seq-TS}}$	21.4
	rand init $+ \mathcal{F}^{\text{state}}_{\text{seq-TS}}$	21.2

The sequence posterior targets used to train the students in Table 11.7 were combined using a sum combination of (6.21). It is also possible to combine the targets using a product combination of (6.23) [78]. As is shown in Section 6.2.3, a product combination over state sequence posteriors is equivalent to a product combination over per-frame acoustic scores. Therefore, the teachers' contribution to the criterion derivative can be computed by performing frame-level combination of the teachers' acoustic scores, then performing a forward-backward operation over a single denominator lattice for the whole ensemble, using these combined acoustic scores. The product combination of the targets is therefore less computationally expensive to use during training than the sum combination, which requires one forward-backward operation for each teacher.

Table 11.8 Sum and product combinations of sequence posterior targets, in AMI-IHM. The students were randomly initialised. There is no significant performance difference between the two target combination methods.

Target combination	Student WER (%)
sum	22.7
product	22.7

The sum and product combinations of the teachers' sequence posterior targets are compared in Table 11.8, using the AMI-IHM dataset. Both students here were trained beginning from random parameter initialisations. The results do not suggest any significant difference between the student performances when using either form of target combinations. It may

therefore be preferable to use a product combination of the sequence posterior targets, to reduce the computational cost when training the student.

11.6.2 State cluster diversity

The sequence-level teacher-student learning experiments in Section 11.6.1 train the student by minimising the KL-divergence between state cluster sequence posteriors. This requires the same support over all of the distributions, and therefore requires that all systems use the same set of state clusters. This limits the allowed forms of diversities that the teacher ensemble can have. Section 6.2.4 instead proposes to train the student, by minimising a KL-divergence between logical context-dependent state sequence posteriors, using the criterion of $\mathcal{F}^{\text{CD}}_{\text{seq-TS}}$ in (6.26). This criterion allows sequence-level teacher-student learning to be performed with a diversity of state cluster sets, while having a similar simplicity and efficiency of computing the criterion derivative as a KL-divergence over state cluster sequence posteriors. A lattice-based implementation of this criterion is assessed in Section 10.2.2. This current section investigates a lattice-free implementation of this criterion.

The first experiment assesses the performance of a lattice-free ensemble that has a diversity of state cluster sets. Ensembles of 4 TDNN-LSTMs were trained, with either a diversity of acoustic model parameters from different random parameter initialisations, or a diversity of state cluster sets using the random forest method, in AMI-IHM. Each ensemble was either trained using a lattice-based implementation of the $\mathcal{F}_{\text{sMBR}}$ criterion in (2.83), or using lattice-free $\mathcal{F}_{\text{MMI}}$. The performances of these ensembles are shown in Table 11.9.

Table 11.9 Comparing ensembles with model parameter and state cluster diversities, using lattice-based and lattice-free training, in AMI-IHM. Each ensemble had 4 TDNN-LSTMs, and was combined using MBR combination decoding. Having different state clusters yields more diversity, but no gain in the combined performance for the lattice-free ensemble.

Training	Ensemble diversity	Single system WER (%) mean	best	worst	std dev	Combined WER (%)	cross-WER (%)
lattice-based $\mathcal{F}_{\text{sMBR}}$	parameter	24.8	24.6	25.0	0.17	22.5	17.2
	state cluster	24.9	24.8	25.0	0.08	22.2	18.6
lattice-free $\mathcal{F}_{\text{MMI}}$	parameter	25.3	25.1	25.4	0.15	22.2	20.8
	state cluster	25.6	25.5	25.6	0.06	22.3	21.3

The results suggest that under both sequence discriminative training frameworks, the ensemble with different sets of state clusters exhibits a greater measured diversity than the ensemble with only different sets of model parameters. However, unlike the lattice-

based ensemble, this greater diversity does not yield any improvements in the combined performance of the lattice-free ensemble with a diversity of state cluster sets.

Table 11.10 Impact of context-dependence, HMM topology, and decision tree size on ensemble diversity for lattice-free systems, in AMI-IHM. Each ensemble had 4 TDNN-LSTMs with different random forest decision trees, and was combined using MBR combination decoding. The lack of combination gain with state cluster diversity is not due to the simplification of the systems.

	Single single WER (%)				Combined	
Ensemble diversity	mean	best	worst	std dev	WER (%)	cross-WER (%)
triphone, 3-state HMM, 4000 decision tree leaves						
parameters	24.8	24.7	24.9	0.10	21.9	19.8
state clusters	25.2	25.1	25.2	0.06	21.9	21.1
biphone, 2-state HMM, 2000 decision tree leaves						
parameters	25.3	25.1	25.4	0.15	22.2	20.8
state clusters	25.6	25.5	25.6	0.06	22.3	21.3

An initial hypothesis for this lack of gains may be because of the use of a simpler HMM topology and context-dependence, and smaller decision trees in the lattice-free systems. These may lead to there being fewer possible permutations to cluster the logical context-dependent states, causing the decision trees in the ensemble to be more similar. Lattice-free ensembles that use a triphone context-dependence, a 3-state HMM, and 4000 decision tree leaves, similar to a lattice-based system, are evaluated in Table 11.10. The 3-state HMM topology shown in Figure 2.2a may not be the most appropriate when used with a lattice-free system that has a 30ms frame shift. Instead, the 3-state HMM topology shown in Figure 11.1 was used to allow a sub-word unit to be traversed in the same minimum time as lattice-based systems [60]. The results suggest that using the more complex system design yields a greater difference between the cross-WERs of the ensembles with parameter and state cluster diversities. However, this again does not lead to any improvement in the combined performance of the ensemble with state cluster diversity, over that of the ensemble with parameter diversity. Regardless of this lack of gains, the intention of this section is to investigate the feasibility of a lattice-free implementation of sequence-level teacher-student learning using the $\mathcal{F}^{\text{CD}}_{\text{seq-TS}}$ criterion, and not to maximise the ensemble diversity and combination gains. It may be possible to improve upon the diversity and combined performance of the ensemble, by explicitly training the decision trees to be different, using the methods proposed in [15, 162]. Applying these together with sequence-level teacher-student learning may be an interesting topic for future research.

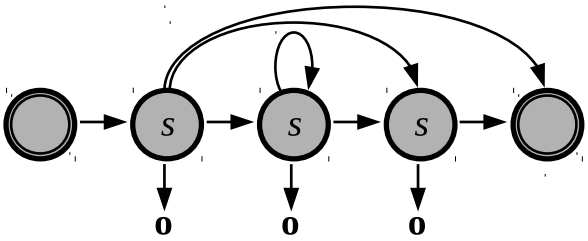


Fig. 11.1 3-state hidden Markov model topology for lattice-free systems. This topology can be traversed with a minimum of one frame.

Although the more complex ensembles perform better, they are also more computationally expensive to use to train a student, especially when different sets of state clusters are used. This is because the criterion derivative is computed with the lattice arcs marked with intersect states, formed using a Cartesian product between the decision trees of the student and all of the teachers. Using the more complex systems leads to there being 118376 intersect states in AMI-IHM, as opposed to having 13838 intersect states with the simpler systems. It can be computationally difficult to implement training over this large number of states. As such, the simpler lattice-free systems were used.

Table 11.11 shows the performance of students trained toward the ensembles with state cluster diversity, using a lattice-free implementation of sequence-level teacher-student learning. The students used the same TDNN-LSTM acoustic model topologies and decision tree sizes as each of the teachers, but with decision trees that were trained with greedy splits. The students were trained from random parameter initialisations. As a reference, the combined ensembles have WERs of 22.3% and 20.6% for AMI-IHM and MGB-3 respectively.

Table 11.11 Lattice-free sequence-level teacher-student learning with different sets of state clusters. TDNN-LSTM students with greedy decision trees were trained toward ensembles of 4 TDNN-LSTM teachers with different random forest decision trees. Training of the students began from random parameter initialisations. The sequence-level student is able to come closer to the combined ensemble performance than can a lattice-free $\mathcal{F}_{\text{MMI}}$ system.

Dataset	Training	WER (%)
AMI-IHM	$\mathcal{F}_{\text{MMI}}$	25.3
	$\mathcal{F}^{\text{CD}}_{\text{seq-TS}}$	23.2
MGB-3	$\mathcal{F}_{\text{MMI}}$	23.6
	$\mathcal{F}^{\text{CD}}_{\text{seq-TS}}$	21.8

The results show that by using a lattice-free implementation of the $\mathcal{F}^{\text{CD}}_{\text{seq-TS}}$ criterion, the students can be trained to perform more closely to the combined ensembles with state

cluster diversity, than can single lattice-free $\mathcal{F}_{\text{MMI}}$ systems. However, these students perform worse than those of the ensembles with parameter diversity in Table 11.7, despite the similar combined performances of the different ensembles. Unlike frame-level teacher-student learning, there is no need to make any approximations when mapping the target posteriors between different sets of state clusters at the sequence level. Therefore, the performance degradation in the students here is not caused by any such target mapping approximation.

Table 11.12 Using larger decision trees for the student, in AMI-IHM. The 2000 and 3000 leaves decision trees used greedy splits, while that with 11581 leaves was constructed using a Cartesian product of the 4 random forest decision trees of the teachers. Increasing the student's decision tree size brings its performance closer to that of the combined ensemble.

Student decision tree size	No. parameters	Student WER (%)
2000	9.6×10^6	23.2
3000	9.9×10^6	23.0
11581 (intersect)	12.1×10^6	22.5

As with frame-level teacher-student learning in Section 9.6.1, the degradation in the student performance may also be due to the limited phonetic resolution of the student, having a standard-sized decision tree, which in AMI-IHM has 2000 leaves. Table 11.12 shows the performances of students with larger decision trees in AMI-IHM. A student using the intersect states as outputs in MGB-3 has a WER of 21.1%. The results here agree with Table 9.11, in suggesting that increasing the student's decision tree size may allow the student performance to come closer to that of the ensemble. This demonstrates the added flexibility of the proposed criterion, in allowing for the freedom to choose the student's decision tree independently of those used by the teachers. Although using a larger decision tree improves the ability of the student to capture the ensemble performance, it also leads to a greater computational cost when performing recognition.

11.7 Multiple forms of diversities

The ensembles used thus far in this thesis have mostly incorporated only a single form of diversity. However, it may be possible to obtain a richer ensemble by using multiple forms of diversity. This section investigates using both acoustic model parameter and state cluster diversities together within an ensemble.

11.7.1 Ensemble from intermediate model iterations

One possible method to simultaneously use diversities in both the model parameters and state cluster sets is to generate multiple decision trees, then train multiple acoustic models for each decision tree, beginning from different parameter initialisations. However, this requires a separate training run for each member of the ensemble, and can therefore be computationally expensive. As is discussed in Section 3.2.4, a cheaper way of incorporating parameter diversity is to use the intermediate model iterations from within each run of training. This is assessed in this section.

The first experiment investigates the diversity that can be gained by using these intermediate model iterations, when the single training run uses different training criteria. Ensembles were constructed out of the intermediate models of the final epoch of training using either the cross-entropy, $\mathcal{F}_{\text{CE}}$, sequence-level lattice-based $\mathcal{F}_{\text{sMBR}}$, or lattice-free $\mathcal{F}_{\text{MMI}}$ criteria, in AMI-IHM. All ensembles used TDNN-LSTM acoustic models and greedy decision trees. To reduce the computational cost when performing combination, the training runs were thinned to use only every 3rd model iteration for the ensembles. The performances of these ensembles are shown in Table 11.13.

Table 11.13 Ensembles generated from intermediate model iterations of single training runs, in AMI-IHM. The TDNN-LSTM topology was used for all systems. Combination was performed using MBR combination decoding. Significant diversity can be obtained by using the intermediate model iterations when performing $\mathcal{F}_{\text{CE}}$ and lattice-free $\mathcal{F}_{\text{MMI}}$ training.

	Single system WER (%)				Combined	
Training	mean	best	worst	std dev	WER (%)	cross-WER (%)
$\mathcal{F}_{\text{CE}}$	26.5	26.3	26.8	0.10	25.4	11.2
lattice-based $\mathcal{F}_{\text{sMBR}}$	24.9	24.7	25.7	0.24	24.7	4.4
lattice-free $\mathcal{F}_{\text{MMI}}$	25.6	25.0	26.5	0.34	23.5	15.6

The results suggest that significant diversity and combination gains can be obtained by using this ensemble generation method, when performing cross-entropy and lattice-free sequence discriminative training. However, when performing lattice-based sequence discriminative training, the combined ensemble performance is not better than that of the best single system. The diversity between these lattice-based systems is very small. This is not due to the $\mathcal{F}_{\text{sMBR}}$ criterion, as an ensemble from a single run of lattice-based $\mathcal{F}_{\text{MMI}}$ training (not shown in Table 11.13) has a best single system WER of 24.9%, combined WER of 25.1%, and cross-WER of 2.7%. The lack of diversity seems to result from the lattice-based training method. Lattice-based sequence discriminative training uses the cross-entropy system as the initial parameters, while the lattice-free method begins from a random

parameter initialisation. This may bias all of the lattice-based model iterations toward the cross-entropy system, limiting the diversity that can be expressed. Furthermore, the limited hypotheses contained within the pruned lattices may limit how much the lattice-based system behaviour is allowed to diverge from that of the cross-entropy system. This demonstrates that another advantage that the lattice-free method has is to allow a diverse sequence-level ensemble to be generated with a single training run.

Table 11.14 Intermediate iterations and random initialisation ensemble methods, in AMI-IHM. All systems used the TDNN-LSTM topology, trained with lattice-free $\mathcal{F}_{\text{MMI}}$. Combination was performed using MBR combination decoding. Significant diversity and combination gains can be obtained from both methods, but more so from using different random parameter initialisations.

	Single system WER (%)				Combined	
Parameter diversity	mean	best	worst	std dev	WER (%)	cross-WER (%)
intermediate iterations	25.6	25.0	26.5	0.34	23.5	15.6
random initialisation	25.3	25.1	25.4	0.15	22.2	20.8

The next experiment compares lattice-free TDNN-LSTM ensembles with model parameter diversity, generated using either multiple training runs from different random parameter initialisations or the intermediate model iterations from a single run of training, in AMI-IHM. The ensemble from a single training run again used every 3rd model iteration from the last epoch of training, leading to 20 systems. The ensemble from different random parameter initialisations had 4 systems. These ensembles are compared in Table 11.14. The results suggest that the diversity and combination gains obtained by using multiple training runs from different random parameter initialisations are greater. The intermediate model iterations may be fairly similar to each other, leading to a limited diversity. However, during training, it is computationally cheaper to perform a single, rather than multiple, training run.

The previous experiments show that model parameter diversity and combination gains can be obtained by simply using the intermediate model iterations from training. This does not add any computational cost when generating the ensemble. The aim of this section is to incorporate multiple forms of diversities into an ensemble. The next experiment generates an ensemble with both state cluster and model parameter diversities, in the AMI-IHM and MGB-3 datasets. Three ensembles were generated. The first had a diversity of model parameters, by using the intermediate model iterations from a single run of training, with a greedy decision tree. The second had a diversity of state cluster sets, by separately training 4 systems, each with a different decision tree, generated using the random forest method. The third incorporated both forms of diversities, by performing 4 training runs with different decision trees, and using the intermediate model iterations from all 4 training runs. In MGB-3,

Table 11.15 Ensembles with both model parameter and state cluster diversities. The ensembles used TDNN-LSTMs, trained with lattice-free $\mathcal{F}_{\text{MMI}}$, and were combined using MBR combination decoding. Parameter diversity used intermediate model iterations from a single run of training, yielding 20 systems for AMI-IHM and 22 systems for MGB-3. State cluster diversity was obtained by performing 4 training runs with different random forest decision trees, and taking the final iteration of each training run. The ensemble with both forms of diversities performed 4 training runs with different random forest decision trees and used the intermediate model iterations from all runs of training, yielding 80 systems for AMI-IHM and 88 systems for MGB-3. Using both forms of diversities together yields a better combined performance.

	Ensemble	Single WER (%)				Combined	cross-
Dataset	diversity	mean	best	worst	std dev	WER (%)	WER (%)
AMI-IHM	parameter	25.6	25.0	26.5	0.34	23.5	15.6
	state cluster	25.6	25.5	25.6	0.06	22.3	21.3
	both	25.9	25.3	27.1	0.34	21.8	20.6
MGB-3	parameter	24.0	23.5	24.7	0.34	20.8	16.9
	state cluster	23.5	23.3	23.7	0.18	20.6	18.2
	both	24.0	23.2	25.2	0.41	20.0	18.6

the intermediate models were taken from every 12th iteration of the final epoch of training, leading to 22 systems per training run. The TDNN-LSTM acoustic model topology was again used, with lattice-free $\mathcal{F}_{\text{MMI}}$ training. The ensemble performances are shown in Table 11.15. The results show that using both forms of diversities together can yield a better combined performance. In AMI-IHM, using either parameter or state cluster diversities alone yields combined WERs of 23.5% and 22.3% respectively, while using both forms of diversities together yields a combined WER of 21.8%. The measured cross-WER diversity may not be large when using both forms of diversities, because its computation includes contributions from pairs of systems within the same training run, which the results also show have smaller cross-WERs.

When using standard training methods, model parameter diversity can be obtained at no additional computational cost, by using the intermediate model iterations of training. It is interesting to question whether this parameter diversity can also be applied to the training run of a student, to further improve upon the student's performance. Following Section 11.6.1, a student was trained using a lattice-free implementation of sequence-level teacher-student learning, toward an ensemble with a diversity of model parameters, generated using multiple training runs from different random parameter initialisations, in AMI-IHM. The student was trained, beginning from a random parameter initialisation. An ensemble of students was generated by taking the intermediate model iterations from the final epoch of training of

the student. Again, every 3rd iteration was used to generate the ensemble of students. This is compared to an ensemble generated from a single run of lattice-free $\mathcal{F}_{\text{MMI}}$ training. The ensemble performances are shown in Table 11.16.

Table 11.16 Ensemble from intermediate student training iterations, in AMI-IHM. The TDNN-LSTM student was trained toward an ensemble of 4 lattice-free $\mathcal{F}_{\text{MMI}}$ TDNN-LSTM teachers from different random parameter initialisations. There is less diversity between the intermediate student iterations.

	Single system WER (%)				Combined	
Training	mean	best	worst	std dev	WER (%)	cross-WER (%)
$\mathcal{F}_{\text{MMI}}$	25.6	25.0	26.5	0.34	23.5	15.6
$\mathcal{F}^{\text{state}}_{\text{seq-TS}}$	22.8	22.5	23.4	0.26	22.3	7.9

The results suggest that the ensemble of students has a smaller diversity, agreeing with the observations in Section 9.4. This may suggest that using soft posterior targets from the teachers may form a smoother criterion error surface, compared to using δ-function hard targets in the $\mathcal{F}_{\text{MMI}}$ criterion, making it easier for the student to learn. This small diversity between the students leads to a small combination gain. This is significantly better than the WER of the final student iteration of 22.7%, with a null hypothesis probability less than 0.001. It may therefore be possible to further leverage upon parameter diversity between the students, after teacher-student learning.

11.7.2 Parameter-level combination

The previous section has shown that model parameter diversity can be generated in a computationally efficient manner by using the intermediate model iterations of a training run. However, this can lead to a large ensemble, which can be computationally expensive to use for recognition, when combining at the hypothesis level. Teacher-student learning can be used to reduce this computational cost. However, this requires a student to be trained to emulate the ensemble, which incurs a computational cost during training. The parameter-level combination method, discussed in Section 4.3.4, can also be used to compress the ensemble into a single model, referred to as the smoothed model. Unlike teacher-student learning, no model training is required to perform parameter-level combination. However, parameter-level combination can only be used on an ensemble where all acoustic models use the same topology and have hidden representations that are similarly ordered. The method of generating an ensemble from the intermediate model iterations of a single training run abides by these

restrictions[1]. Furthermore, Section 11.7.1 suggests that this ensemble generation method can yield a diverse ensemble. This current section investigates different approaches to compress this form of ensemble, to reduce the computational cost when performing recognition.

Table 11.17 Methods of combining an ensemble of intermediate model iterations. The models were obtained from the last epoch of lattice-free $\mathcal{F}_{\text{MMI}}$ training of TDNN-LSTMs. The sequence-level students used the same TDNN-LSTM topology.

Dataset	Mean single WER (%)	Combined WER (%) hypothesis	parameter	student
AMI-IHM	25.6	23.5	23.8	22.6
MGB-3	24.0	20.8	21.3	21.3

In the AMI-IHM and MGB-3 datasets, ensembles of lattice-free TDNN-LSTMs were each generated by taking the intermediate model iterations from a single training run. These were combined using either hypothesis-level MBR combination decoding of (3.29), parameter-level combination of (4.26), or by training a student toward the ensemble using sequence-level teacher-student learning. When performing parameter-level combination, preliminary experiments suggested that no significant performance gains were obtained when explicitly training the interpolation weights toward the $\mathcal{F}_{\text{MMI}}$ criterion. Therefore, the simple setup of equal interpolation weights was used. The combined ensemble performances are shown in Table 11.17. The results suggest that hypothesis-level combination performs better than parameter-level combination for both datasets, with null hypothesis probabilities less than 0.001. In AMI-IHM, the student gives the best performance of all the combination methods. However, no consistent performance gain is observed for the student in MGB-3.

Both parameter-level combination and teacher-student learning compress the ensemble into a single compressed system. As is shown in Table 11.17, these compressed systems can perform better than the constituent systems in the ensemble. The parameter-level combination method can do this without any additional computational cost.

The combined performance of an ensemble depends on both the individual system performances and the diversity between the systems. Parameter-level combination can be used to produce smoothed models with good performances. Multiple smoothed models can be used to construct an ensemble. The next experiment investigates an ensemble of these smoothed models. Two ensembles of lattice-free TDNN-LSTM systems were generated in each of the AMI-IHM and MGB-3 datasets. Each ensemble consisted of 4 systems, each having a different decision tree, obtained using the random forest method. The members of

[1] It is a standard implementation in the Kaldi training recipes to perform parameter-level combination over the intermediate model iterations within the final epoch of training.

the ensembles were either taken from the last training iteration, or taken as the smoothed models, obtained by performing parameter-level combination across the intermediate model iterations in the final epoch of each training run. The ensembles were combined using hypothesis-level MBR combination decoding. The ensemble performances are shown in Table 11.18.

Table 11.18 Ensembles made of models from the last iteration of training or from smoothed models. All ensembles used 4 lattice-free $\mathcal{F}_{\text{MMI}}$ TDNN-LSTM systems, and were combined using MBR combination decoding. Using smoothed models in the ensemble yields less diversity, but better individual system performances, leading to a better combined performance.

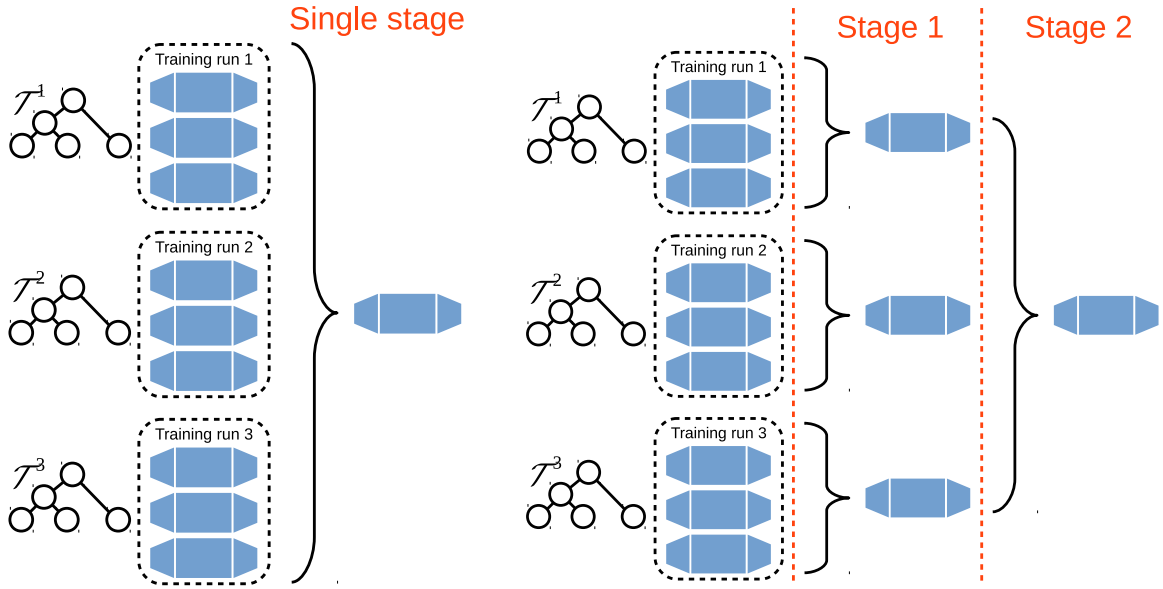
Dataset	Ensemble members	Single system WER (%) mean	best	worst	std dev	Combined WER (%)	cross-WER (%)
AMI-IHM	last iteration	25.6	25.5	25.6	0.06	22.3	21.3
	smoothed	24.2	24.1	24.2	0.06	21.6	18.2
MGB-3	last iteration	23.5	23.3	23.7	0.18	20.6	18.2
	smoothed	21.3	21.2	21.4	0.08	19.7	12.8

The results show that using the smoothed models from each training run as members of the ensemble yields better individual system performances. However, this also leads to a reduction in the diversity between the systems. The better single system performances yield an overall better combined performance, when using the smoothed models.

11.7.3 Multi-stage compression

The results in Table 11.15 suggest that using multiple forms of diversities within an ensemble can yield a good combined performance. In this, both state cluster and model parameter diversities were obtained by using multiple training runs with different decision trees, and taking the intermediate model iterations from all training runs. However, this can lead to a large ensemble that can be computationally expensive to use for recognition. Sequence-level teacher-student learning can be used to compress the ensemble into a single student. The ensemble of smoothed models in Table 11.18 is another possible method to reduce this computational cost. Here, parameter-level combination is first performed over each training run, to produce a smaller collection of smoothed models, which are then combined. This represents a 2-stage combination process. It is also possible to train a student toward the ensemble of smoothed models.

When compressing the whole ensemble into a single student, it is interesting to consider whether compressing in multiple stages, in Figure 11.2b, may be better than training the student toward all of the intermediate model iterations of all of the training runs, in Figure



(a) Training a student toward all systems, from the intermediate iterations of the multiple training runs.

(b) 2-stage compression, by compressing within each training run, then compressing across the multiple training runs.

Fig. 11.2 Multi-stage compression of an ensemble that uses multiple training runs and the intermediate models from within each training run. Each training run uses a different decision tree.

11.2a. This is also motivated by the results in Table 11.17, which suggest that different combination methods across the intermediate model iterations of a single training run may perform differently. This section investigates a 2-stage compression scheme, of first combining within each of the separate training runs, using either parameter-level combination or teacher-student learning. The smoothed models or students across the different training runs are then compressed into a single student. It is difficult to use parameter-level combination across the different training runs, as the systems are not constrained to have hidden representations that are similarly ordered and have different sets of state clusters.

In the AMI-IHM and MGB-3 datasets, ensembles of lattice-free TDNN-LSTMs were generated with both state cluster and model parameter diversities, by using multiple training runs with different decision trees, and using the intermediate model iterations from each training run. The multiple decision trees were again generated using the random forest method. The different compression schemes are shown in Table 11.19.

The results suggest that the best final student can be obtained by first performing parameter-level combination within each training run, then performing teacher-student learning toward these smoothed models across the multiple training runs. These final students have WERs of 22.7% and 21.2% for AMI-IHM and MGB-3 respectively. However,

Table 11.19 Multi-stage compression methods. The ensembles used the intermediate model iterations of 4 training runs of lattice-free $\mathcal{F}_{\text{MMI}}$ TDNN-LSTMs with different random forest decision trees. The students used the same TDNN-LSTM topology as each teacher. The best stage 2 student performance is obtained by first performing stage 1 parameter-level combination.

Dataset	Stage 1 compression	Stage 1 WER (%) mean	Stage 1 WER (%) std dev	Stage 1 cross-WER (%)	Stage 2 WER (%) hypothesis	Stage 2 WER (%) student
AMI-IHM	-	25.9	0.34	20.6	21.8	22.9
	parameter	24.2	0.06	18.2	21.6	22.7
	student	22.8	0.10	14.1	21.3	23.1
MGB-3	-	24.0	0.41	18.6	20.0	21.5
	parameter	21.3	0.08	12.8	19.7	21.2
	student	21.3	0.06	10.0	20.1	21.9

these students are not significantly better than students trained directly toward all of the intermediate model iterations of all of the training runs, without a stage 1 compression, having WERs of 22.9% and 21.5% for AMI-IHM and MGB-3 respectively, with null hypothesis probabilities of 0.204 and 0.016. Despite this, there is a consistent improvement across both datasets. This 2-stage compression also has the advantage of being less computationally expensive, as this trains a student toward fewer teachers. Parameter-level combination has negligible computational cost, as the interpolation weights used here were not trained.

Comparing the stage 1 compression methods, performing teacher-student learning within each training run yields a better performance than parameter-level combination in AHI-IHM, but this trend is not replicated in MGB-3, similarly to the results in Table 11.17. A stage 1 compression using teacher-student learning leads to less diversity between the students of the different training runs with a cross-WER of 14.1% in AMI-IHM, than that between the smoothed models of parameter-level combination with a cross-WER of 18.2%. The small diversity between students agrees with the observations in Section 9.4. This lack of diversity may be a reason why training a final student toward the stage 1 students, yielding a WER of 23.1% in AMI-IHM, does not perform as well as training a final student toward the stage 1 parameter-level combinations, with a WER of 22.7%.

Although the stage 2 hypothesis-level combinations are able to gain from both forms of diversities, shown in Table 11.15, the best stage 2 students are not able to significantly outperform the stage 1 students. In AMI-IHM, the best stage 2 student has a WER of 22.7%, while the mean stage 1 student WER is 22.8%. The stage 1 students used the same sets of state clusters as their teachers, while the stage 2 students were trained toward teachers with different sets of state clusters. Similarly to the observation in Section 11.6.2, the stage 2

students here may again be limited by their phonetic resolutions, as these students used the same-sized decision trees as each of the teachers.

Table 11.20 Improving the phonetic resolution of stage 2 students. The students with 2000 and 3600 leaves used greedy splits. The 11581 and 17236 intersect states were constructed using the Cartesian products of the 4 random forest decision trees of the teachers. The multi-task systems used the same 4 decision trees as the teachers. Using both the multi-task topology and a single large output layer brings the student performance closer to that of the combined ensemble.

Dataset	Student	No. parameters	Student WER (%)
AMI-IHM	2000 state clusters	9.6×10^6	22.7
	11581 intersect states	12.1×10^6	22.1
	multi-task	11.2×10^6	22.2
MGB-3	3600 state clusters	10.1×10^6	21.2
	17236 intersect states	13.6×10^6	20.8
	multi-task	12.9×10^6	20.6

The performances of stage 2 students that used the intersect states are shown in Table 11.20. These students were trained toward the stage 1 smoothed models. As a reference, the combined WERs of these teacher ensembles are 21.6% and 19.7% for AMI-IHM and MGB-3 respectively, from Table 11.19. The results show that students that use intersect states are able to come closer to the teacher ensemble performances. However, these intersect state students have many model parameters, and may therefore be computationally expensive to use when performing recognition. A multi-task topology can instead be used as the final stage 2 student acoustic model, as is discussed in Section 5.1.2. Frame-level combination is used with the multi-task student, as it is less computationally expensive than hypothesis-level combination. The frame-level combination is also taken into account during sequence-level teacher-student learning, by back-propagating the criterion derivative through it, as is described in Section 4.2.1. Table 11.20 suggests that the multi-task stage 2 students are able to perform similarly to the students with single large output layers of intersect states, but have fewer model parameters. These results suggest that the students can benefit from the diversities provided by having different sets of state clusters and different sets of model parameters.

11.8 Summary

This chapter has investigated the ensemble generation and teacher-student learning methods applied to lattice-free training. The single lattice-free systems are able to perform competitively with lattice-based systems, while the lattice-free ensembles are able to exhibit greater

diversity. The experiments have demonstrated that the proposed frame and sequence-level teacher-student learning methods can be used in a lattice-free implementation. Finally, state cluster and model parameter diversities can be used together to generate a rich ensemble. This ensemble can be efficiently and effectively compressed into a single student using a 2-stage compression scheme.

Chapter 12

Conclusion

This thesis has investigated approaches to generate and compress a diverse ensemble of multiple systems. The combined ensemble performance depends on both the accuracy of the individual systems and the diversity between the system behaviours. Many possible forms of diversities can be used in ASR. Although an ensemble may perform well, it can be computationally expensive to use for recognition. Teacher-student learning is one possible method that can be used to compress an ensemble. The standard method trains the student to emulate the ensemble by propagating information about the per-frame state cluster posteriors. This is limited in two ways. First, it requires that all systems in the ensemble use the same set of state clusters, which limits the forms of diversities that the ensemble is allowed to have. Second, frame-level posterior information may not effectively convey the sequential nature of speech data. This thesis has proposed several generalisations of the teacher-student learning framework, to overcome these limitations. This final chapter provides a detailed summary of this thesis, and presents several possible directions for future research.

12.1 Summary

The first contribution of this thesis is to compare approaches of generating diverse ensembles, discussed in Chapter 3. Section 3.3 has described how in ASR, many possible forms of diversities can be used. Using more forms of diversities may allow for a richer ensemble. Of these forms, experiments in Chapter 8 suggest that significant combination gains can be obtained by introducing diversity into the acoustic model parameters and topology, set of state clusters, and set of sub-word units. When performing recognition, the members of the ensemble need to be combined together. As has been described in Section 3.4, hypothesis-level combination is the most computationally expensive approach, as it requires multiple decoding runs. Frame-level combination only a requires a single decoding run, but

data still needs to be fed through each of the acoustic models. Feature-level combination only requires data to be fed through a single acoustic model classifier, but only allows the ensemble to encompass a diversity in the feature representations. In Section 3.5 the ESN has been proposed as a method of obtaining feature diversity, by randomly sampling feature representations. This ensemble method can readily be used with feature-level combination for computational efficiency. However, experiments in Section 8.2 do not seem to indicate that this ensemble method can provide any significant performance gains.

Although an ensemble may perform well, it can be computationally expensive to use for recognition. Chapter 4 has reviewed several approaches to compress an ensemble, to reduce this computational cost. One possible method is teacher-student learning, which compresses the ensemble into a single student, such that only the student needs to be used to perform recognition. The standard form of teacher-student learning, described in Section 4.3.3, trains the student to emulate the combined ensemble of teachers, by propagating information from the teachers to the student in the form of per-frame state cluster posteriors. Experiments in Chapter 9 have demonstrated the ability to train a student using this method, with the student achieving a WER of 25.1%, which is closer to the WER of the combined ensemble of 24.9%, than the WER that can be achieved by a standard sequence-trained system of 25.7%, in AMI-IHM. The results of these experiments suggest that it is the information about how difficult the teachers believe that each frame is to classify, that is useful to the student.

However, standard frame-level teacher-student learning is limited in two ways. First, all systems in the ensemble are required to use the same set of state clusters. This limits the forms of diversity that the ensemble is allowed to have. Second, the frame-level information that is propagated may not effectively convey all aspects about the sequential nature of speech data. This thesis has proposed extensions to the teacher-student learning framework to overcome these limitations.

The second contribution of this thesis addresses the first limitation, by generalising the frame-level teacher-student learning framework to allow for different sets of state clusters. The proposed method, discussed in Section 5.1.1, minimises the KL-divergence between logical context-dependent state cluster posteriors, and leads to an approximate method of mapping posteriors between different sets of state clusters. Experiments in Section 9.6.1 demonstrate that this can train a student to emulate an ensemble with a diversity of state cluster sets. The student here is able to achieve a WER of 25.5%, which comes closer to the combined ensemble WER of 24.5%, than a standard cross-entropy system with a WER of 28.4%, in AMI-IHM. However, the results also suggest that the student may require a large set of state clusters to effectively emulate the ensemble. In the 207V dataset, a student with a large output layer of intersect states has a WER of 46.6%, which is better than a student with

a standard-sized output layer with a WER of 47.3%. The proposed criterion allows for the freedom to choose the size of the student's set of state clusters, independently of those used within the ensemble. However, using a large set of state clusters may require many model parameters, which may impose a high computational cost when performing recognition.

The multi-task ensemble, discussed in Section 4.2, is an alternative method of compressing an ensemble with a diversity of state cluster sets. This ties together the parameters of the hidden layers across all members of the ensemble, leaving only separate output layers for each set of state clusters. As such, data only needs to be fed through the hidden layers once for the whole ensemble. However, experiments in Section 8.7 suggest that the sharing of parameters between the members of the multi-task ensemble causes a reduction in the diversity, and results in smaller combination gains. In AMI-IHM, a sequence-trained multi-task ensemble has a cross-WER of 11.7% and combined WER of 25.5%, which is worse than an ensemble of separate systems, having a cross-WER of 15.2% and combined WER of 24.6%, even though the individual ensemble members perform similarly. Section 5.1.2 has proposed to use teacher-student learning to allow the multi-task ensemble to learn from the diverse behaviours of separate systems. The experiments in Section 9.6.2 suggest that this can yield a better multi-task ensemble performance, with a combined WER of 24.4% in AMI-IHM. Furthermore, the multi-task ensemble can outperform a student with a single large output layer, while having fewer model parameters, thereby representing a better balance between the number of model parameters and output complexity.

The third contribution of this thesis addresses the second limitation of standard teacher-student learning, by considering two alternative forms of information that can be propagated to the student. These have been discussed in Chapter 6. Section 6.2 has generalised teacher-student learning to the sequence level, by propagating sequence posterior information from the teachers to the student. This directly conveys information about the sequence-level behaviours of the teachers, which may be useful in the sequence modelling task of ASR. One possible criterion is to minimise the KL-divergence between word sequence posteriors. This places few restrictions on the forms of diversities that are allowed within the ensemble, but has a derivative that can be expensive to compute when training the student. An alternative criterion of a KL-divergence between lattice arc sequence posteriors has been proposed in Section 6.2.2, which forms an upper bound to the word sequence KL-divergence. Its derivative can be computed using two levels of forward-backward operations over lattices representing the competing hypotheses. Section 6.2.3 has shown that by marking the arcs with state clusters, the criterion derivative can be further simplified to only require a single level of forward-backward operations. Experiments in Section 10.2.1 suggest that propagating hypothesis posterior information using this proposed criterion can yield a better

student performance than when using frame-level teacher-student learning. In AMI-IHM, the sequence-level student achieves a WER of 24.7%, which is better than the frame-level student WER of 25.1%. However, this form of arc marking requires that all systems must again use the same set of state clusters, which limits the allowed forms of ensemble diversities. Section 6.2.4 has proposed to instead mark the arcs with logical context-dependent states. This allows the set of state clusters to differ between systems, while having a criterion derivative that can still be computed simply by using a single level of forward-backward operations. Furthermore, unlike frame-level training, this sequence-level criterion does not require any approximations to be made when mapping sequence posterior targets between different sets of state clusters. The experiments in Section 10.2.2 demonstrate that this can effectively train a student to emulate an ensemble that has a diversity of state cluster sets, with the sequence-level student here achieving a WER of 24.6%, which is again better than the frame-level student WER of 25.5%.

Section 6.1 has also discussed the possibility of propagating information about the hidden layer representations. A KL-divergence criterion has been proposed to propagate information about the discriminability of the hidden layer representations. The experiments in Section 10.1 suggest that such information may allow the student to develop better hidden representations, which aid in further sequence discriminative training.

The proposed methods have also been assessed using lattice-free sequence training. In the lattice-free method, no initial acoustic scores are required to prune lattices, and therefore sequence discriminative training can begin from a random parameter initialisation. The experiments in Section 11.3 suggest that because of this, the lattice-free systems do not suffer from a common bias toward cross-entropy forced alignments, and are therefore able to develop more diverse behaviours. However, the lattice-free acoustic models are often designed to directly produce log-acoustic scores, instead of state cluster posteriors, since initial cross-entropy training is not required. As such, frame-level teacher-student learning needs to be modified to be used with such systems. Section 5.2 has discussed possible modifications, and a KL-divergence style criterion has been proposed. The experiments in Section 11.5 demonstrate that this proposed modification can effectively train a student to learn from lattice-free teachers, with the student WER of 22.9% being closer to the combined ensemble WER of 22.2%, than a standard lattice-free system with a WER of 25.3%, in AMI-IHM. Rather than propagating frame-level information, information about the sequence-level behaviours of the teachers can also be propagated. The experiments in Section 11.6 demonstrate that the proposed sequence-level teacher-student learning methods can be implemented within a lattice-free framework, with the sequence-level student achieving a WER of 22.7%, in AMI-IHM.

Using multiple forms of diversities may allow for a richer ensemble. The experiments in Section 11.7 have investigated using an ensemble with both acoustic model parameter and state cluster diversities. Diversity in the model parameters can be obtained in a computationally efficient manner by using the intermediate model iterations in each run of training. However, using these multiple forms of diversities can lead to a large ensemble, which can be computationally expensive to use for recognition. This ensemble can be effectively compressed into a single student using a 2-stage scheme, by first performing parameter-level combination within each training run, then training the student toward the smoothed models. Using this 2-stage compression scheme and a multi-task topology for the student, the student is able to achieve a WER of 22.2%, which is close to the combined ensemble WER of 21.6%, in AMI-IHM.

12.2 Future work

This thesis has considered generalisations of the teacher-student learning framework, to allow for more forms of ensemble diversities. There is still much scope for further innovation along this aim. The proposed frame-level teacher-student learning criterion in (5.5) allows different sets of state clusters to be used. In this method, the target posteriors are obtained through (5.14), using the approximate mapping of $P\left(s^{\boldsymbol{\Theta}}\middle|s^{m}\right)$ to transform the posteriors from the teachers' set of state clusters to that of the student. This thesis has proposed to estimate $P\left(s^{\boldsymbol{\Theta}}\middle|s^{m}\right)$ from forced logical context-dependent state alignments. It may be possible to further generalise this to allow for diversity within not only the set of state clusters, but also the HMM topology and set of sub-word units. All that is required to obtain target posteriors to train the student is the estimation of an appropriate $P\left(s^{\boldsymbol{\Theta}}\middle|s^{m}\right)$ map. For either of these additional forms of diversities, it may still be possible to estimate this map from a pair of forced alignments.

This thesis has proposed sequence-level teacher-student learning, with lattice arcs marked with either state clusters or logical context-dependent states, for simple and efficient criteria derivative computations. However, these arc markings still require that the members of the ensemble have the same HMM topology and set of sub-word units as the student. Marking the arcs with words removes these restrictions. However, in a lattice-free framework, graphs are often only composed up to the phone level, to limit the computational cost when performing training. It may be interesting to investigate how the lattice-free framework can be adapted to allow for sequence-level teacher-student learning with arcs marked with words. This may importantly allow for sequence posterior information to be propagated between systems with completely different architectures.

However, even if the arcs are marked with words, a KL-divergence between arc sequence posteriors is still only an upper bound to the KL-divergence between word sequence posteriors. The start and end times of each arc are known. This therefore requires that the frame shifts used by the student and teachers must be the same, for the arcs to overlap. This may limit the diversity that a teacher ensemble is allowed to have. However, the derivative of the more general KL-divergence criterion between word sequence posteriors is expensive to compute, because of the need to sum over all possible hypotheses. One possible solution may be to approximate the derivative using Monte Carlo sampling, in a similar manner to the method proposed in [132]. It can be noticed that the criterion in (6.12) can be expressed as an expectation of log-posteriors,

$$\mathcal{F}^{\text{word}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right) = -\mathbb{E}_{\boldsymbol{\omega}\sim P\left(\boldsymbol{\omega}\middle|\mathbf{O}_{1:T},\widehat{\boldsymbol{\Phi}}\right)}\left\{\log P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T},\boldsymbol{\Theta}\right)\right\}. \tag{12.1}$$

The derivative can then be expressed as

$$\frac{\partial \mathcal{F}^{\text{word}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right)}{\partial z^{(K+1)}_{st}} = \kappa\left[P\left(s_t = s|\mathbf{O}_{1:T},\boldsymbol{\Theta}\right) - \mathbb{E}_{\boldsymbol{\omega}\sim P\left(\boldsymbol{\omega}\middle|\mathbf{O}_{1:T},\widehat{\boldsymbol{\Phi}}\right)}\left\{P\left(s_t = s|\boldsymbol{\omega},\mathbf{O}_{1:T},\boldsymbol{\Theta}\right)\right\}\right]. \tag{12.2}$$

The computational cost of computing this expectation can be reduced by using a Monte Carlo approximation with a finite number of hypothesis samples.

The sequence-level teacher-student learning criteria proposed in this thesis propagate information in the form of sequence posteriors. These criteria allow the student to emulate the behaviours of the teachers, under the assumption that MAP decoding is used for recognition. However, performance gains can often be obtained by using MBR decoding. It may therefore be better to propagate information about the expected risks, using a criterion of the form of (6.8). It may be interesting to investigate appropriate forms of distance measures that can be used to propagate this information.

The experiments presented in this thesis suggest that it is important to carefully design the student to be able to effectively emulate the teachers. The experiments suggest that a DNN student may not learn well from a BLSTM teacher, and also that a student with a small set of state clusters may not be able to effectively emulate an ensemble with a diversity of state cluster sets. It may be interesting to investigate what the rules are that govern what form of student is able to learn from what form of teacher. Such an investigation may be particularly useful when applied to the area of domain adaptation [75, 92]. In these scenarios, the teacher often has access to capabilities or information that the student is lacking. Knowing the rules may therefore be useful in designing appropriate students.

Appendix A

Relation between arc and word sequence criteria

Two forms of criteria are proposed in this thesis to perform sequence-level teacher-student learning. This first minimises the KL-divergence between word sequence posteriors,

$$\mathcal{F}^{\text{word}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right) = -\sum_{\boldsymbol{\omega}} P\left(\boldsymbol{\omega}\middle|\mathbf{O}_{1:T}, \hat{\boldsymbol{\Phi}}\right) \log P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right), \tag{A.1}$$

where $\boldsymbol{\Theta}$ and $\hat{\boldsymbol{\Phi}}$ are the student and teacher ensemble respectively, $\boldsymbol{\omega}$ are the word sequence hypotheses, and $\mathbf{O}_{1:T}$ are the observation sequences, where T is the utterance length. However, as is described in Section 6.2.1, the derivative of this criterion can be computationally expensive to compute. A second proposed criterion is to minimise the KL-divergence between lattice arc sequence posteriors,

$$\mathcal{F}^{\text{arc}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right) = -\sum_{\boldsymbol{\omega}} \sum_{\mathbf{a}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega}\middle|\mathbf{O}_{1:T}, \hat{\boldsymbol{\Phi}}\right) \log P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right), \tag{A.2}$$

where $\mathbf{a}_{1:T}$ are the lattice arc sequences and $\mathcal{G}_{\boldsymbol{\omega}}$ is the set of arc sequences that can represent the word sequence $\boldsymbol{\omega}$. The lattice arcs can be marked with words, sub-word units, or states, leading to KL-divergence criteria between the respective acoustic unit sequences. The word and arc sequence posteriors are related as

$$P\left(\boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) = \sum_{\mathbf{a}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega}|\mathbf{O}_{1:T}, \boldsymbol{\Theta}\right). \tag{A.3}$$

As is discussed in Section 6.2.2, the derivative of the $\mathcal{F}^{\text{arc}}_{\text{seq-TS}}$ criterion is less computationally expensive to compute.

The $\mathcal{F}^{\text{arc}}_{\text{seq-TS}}$ criterion is an upper bound to $\mathcal{F}^{\text{word}}_{\text{seq-TS}}$. To prove this, the $\mathcal{F}^{\text{word}}_{\text{seq-TS}}$ criterion in (A.1) can be expressed as

$$\mathcal{F}^{\text{word}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right) = -\sum_{\boldsymbol{\omega}} \sum_{\mathbf{a}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \hat{\boldsymbol{\Phi}}\right) \log \left[\sum_{\mathbf{a}'_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{a}'_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) \right]. \tag{A.4}$$

Being valid probability distributions, the arc sequence posteriors satisfy

$$0 \leq P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) \leq 1 \quad \text{and} \quad \sum_{\mathbf{a}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) \leq 1. \tag{A.5}$$

From these two relationships, it can be seen that $\forall \mathbf{a}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}$,

$$1 \geq \sum_{\mathbf{a}'_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{a}'_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) \geq P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) \tag{A.6}$$

$$0 \leq -\log \left[\sum_{\mathbf{a}'_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{a}'_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) \right] \leq -\log P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right). \tag{A.7}$$

Substituting this relationship into (A.4) leads to

$$\mathcal{F}^{\text{word}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right) \leq -\sum_{\boldsymbol{\omega}} \sum_{\mathbf{a}_{1:T} \in \mathcal{G}_{\boldsymbol{\omega}}} P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega} \middle| \mathbf{O}_{1:T}, \hat{\boldsymbol{\Phi}}\right) \log P\left(\mathbf{a}_{1:T}, \boldsymbol{\omega} | \mathbf{O}_{1:T}, \boldsymbol{\Theta}\right) \tag{A.8}$$

$$\leq \mathcal{F}^{\text{arc}}_{\text{seq-TS}}\left(\boldsymbol{\Theta}\right). \tag{A.9}$$

Minimising $\mathcal{F}^{\text{arc}}_{\text{seq-TS}}$ therefore minimises an upper bound to $\mathcal{F}^{\text{word}}_{\text{seq-TS}}$.

References

[1] Bahl, L. R., Brown, P. F., de Souza, P. V., and Mercer, R. L. (1986). Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *ICASSP*, pages 49–52, Tokyo, Japan.

[2] Bahl, L. R., Jelinek, F., and Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190.

[3] Baker, J. K. (1975). The DRAGON system: an overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):24–29.

[4] Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In *Symposium on Inequalities*, pages 1–8, Los Angeles, USA.

[5] Bell, P. (2017). MGB challenge. http://www.mgb-challenge.org/english.html.

[6] Bell, P., Gales, M. J. F., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Wester, M., and Woodland, P. C. (2015). The MGB challenge: evaluating multi-genre broadcast media recognition. In *ASRU*, pages 687–693, Scottsdale, USA.

[7] Bell, P., Swietojanski, P., and Renals, S. (2017). Multitask learning of context-dependent targets in deep neural network acoustic models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(2):238–247.

[8] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

[9] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2006). Greedy layer-wise training of deep networks. In *NIPS*, pages 153–160, Vancouver, Canada.

[10] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer Science + Business Media.

[11] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural networks. In *ICML*, pages 1613–1622, Lille, France.

[12] Bourlard, H. A. and Morgan, N. (1994). *Connectionist speech recognition: a hybrid approach*. Kluwer Academic Publishers.

[13] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

[14] Breslin, C. (2008). *Generation and combination of complementary systems for automatic speech recognition*. PhD thesis, University of Cambridge.

[15] Breslin, C. and Gales, M. J. F. (2007). Complementary system generation using directed decision trees. In *ICASSP*, pages 337–340, Honolulu, USA.

[16] Brown, G., Wyatt, J., Harris, R., and Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20.

[17] Bucilă, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *KDD*, pages 535–541, Philadelphia, USA.

[18] Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., and Wellner, P. (2005). The AMI meeting corpus: a pre-announcement. In *MLMI*, pages 28–39, Edinburgh, UK.

[19] Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1):41–75.

[20] Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: a neural network for large vocabulary conversational speech recognition. In *ICASSP*, pages 4960–4964, Shanghai, China.

[21] Chan, W., Ke, N. R., and Lane, I. (2015). Transferring knowledge from a RNN to a DNN. In *Interspeech*, pages 3264–3268, Dresden, Germany.

[22] Chen, D., Mak, B., Leung, C.-C., and Sivadas, S. (2014a). Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition. In *ICASSP*, pages 5592–5596, Florence, Italy.

[23] Chen, H., Lundberg, S., and Lee, S.-I. (2017). Checkpoint ensembles: ensemble methods from a single training process. arXiv preprint arXiv:1710.03282.

[24] Chen, T., Fox, E. B., and Guestrin, C. (2014b). Stochastic gradient Hamiltonian Monte Carlo. In *ICML*, pages 1683–1691, Beijing, China.

[25] Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. (2015). The loss surfaces of multilayer networks. In *AISTATS*, pages 192–204, San Diego, USA.

[26] Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 14(3):326–334.

[27] Cui, J., Kingsbury, B., Ramabhadran, B., Saon, G., Sercu, T., Audhkhasi, K., Sethy, A., Nussbaum-Thom, M., and Rosenberg, A. (2017). Knowledge distillation across ensembles of multilingual models for low-resource languages. In *ICASSP*, pages 4825–4829, New Orleans, USA.

[28] Cui, J., Kingsbury, B., Ramabhadran, B., Sethy, A., Audhkhasi, K., Cui, X., Kislal, E., Mangu, L., Nussbaum-Thom, M., Picheny, M., Tüske, Z., Golik, P., Schlüter, R., Ney, H., Gales, M. J. F., Knill, K. M., Ragni, A., Wang, H., and Woodland, P. C. (2015). Multilingual representations for low resource speech recognition and keyword search. In *ASRU*, pages 259–266, Scottsdale, USA.

[29] Damper, R. I. (1995). Self-learning and connectionist approaches to text-phoneme conversion. In Levy, J., Bairaktaris, D., Bullinaria, J., and Cairns, P., editors, *Connectionist models of memory and language*, pages 117–144. UCL Press.

[30] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *NIPS*, pages 2933–2941, Montreal, Canada.

[31] Davis, K. H., Biddulph, R., and Balashek, S. (1952). Automatic recognition of spoken digits. *Journal of the Acoustical Society of America*, 24(6):637–642.

[32] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

[33] Deng, L. and Platt, J. C. (2014). Ensemble deep learning for speech recognition. In *Interspeech*, pages 1915–1919, Singapore.

[34] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157.

[35] Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. (2014). Bayesian sampling using stochastic gradient thermostats. In *NIPS*, pages 3203–3211, Montréal, Canada.

[36] Dupont, S., Ris, C., Deroo, O., and Poitoux, S. (2005). Feature extraction and acoustic modeling: an approach for improved generalization across languages and accents. In *ASRU*, pages 29–34, San Juan, USA.

[37] Evermann, G. and Woodland, P. C. (2000). Posterior probability decoding, confidence estimation and system combination. In *Speech Transcription Workshop*, College Park, USA.

[38] Fiscus, J. G. (1997). A post-processing system to yield reduced word error rates: recognizer output voting error reduction (ROVER). In *ASRU*, pages 347–354, Santa Barbara, USA.

[39] Fiscus, J. G. (1998). 1997 English broadcast news speech (HUB4) LDC98S71. https://catalog.ldc.upenn.edu/LDC98S71.

[40] Fitt, S. and Richmond, K. (2006). Redundancy and productivity in the speech technology lexicon - can we do better? In *ICSLP*, pages 165–168, Pittsburgh, USA.

[41] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

[42] Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *ICML*, pages 1050–1059, New York, USA.

[43] Gales, M. J. F. (1998). Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12(2):75–98.

[44] Gales, M. J. F., Kim, D. Y., Woodland, P. C., Chan, H. Y., Mrva, D., Sinha, R., and Tranter, S. E. (2006). Progress in the CU-HTK broadcast news transcription system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1513–1525.

[45] Gales, M. J. F., Knill, K. M., and Ragni, A. (2015). Unicode-based graphemic systems for limited resource languages. In *ICASSP*, pages 5186–5190, Brisbane, Australia.

[46] Gauvain, J. L., Lamel, L. F., Adda, G., and Adda-Decker, M. (1994). The LIMSI Nov93 WSJ system. In *ARPA Spoken Language Technology Workshop*.

[47] Gemello, R., Mana, F., Scanzio, S., Laface, P., and De Mori, R. (2006). Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training. In *ICASSP*, pages 1189–1192, Toulouse, France.

[48] Gibson, M. and Hain, T. (2006). Hypothesis spaces for minimum Bayes risk training in large vocabulary speech recognition. In *Interspeech*, pages 2406–2409, Pittsburgh, USA.

[49] Gillick, L. and Cox, S. J. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *ICASSP*, pages 532–535, Glasgow, UK.

[50] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, Sardinia, Italy.

[51] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *AISTATS*, pages 315–323, Fort Lauderdale, USA.

[52] Gopalakrishnan, P. S., Kanevsky, D., Nádas, A., and Nahamoo, D. (1989). A generalization of the Baum algorithm to rational objective functions. In *ICASSP*, pages 631–634, Glasgow, UK.

[53] Graff, D. (1997). 1996 English broadcast news speech (HUB4) LDC97S44. https://catalog.ldc.upenn.edu/LDC97S44.

[54] Graves, A. (2012). Sequence transduction with recurrent neural networks. In *ICML Representation Learning Workshop*, Edinburgh, UK.

[55] Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, Pittsburgh, USA.

[56] Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, pages 1764–1772, Beijing, China.

[57] Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. In *IJCNN*, pages 2047–2052, Montreal, Canada.

[58] Grézl, F., Karafiát, M., Kontár, S., and Černocký, J. (2007). Probabilistic and bottle-neck features for LVCSR of meetings. In *ICASSP*, pages 757–760, Honolulu, USA.

[59] Gutmann, M. U. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361.

[60] Hadian, H., Sameti, H., Povey, D., and Khudanpur, S. (2018). End-to-end speech recognition using lattice-free MMI. In *Interspeech*, pages 12–16, Hyderabad, India.

[61] Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

[62] Harper, M. P. (2011). IARPA babel program. http://www.iarpa.gov/index.php/research-programs/babel.

[63] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

[64] Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752.

[65] Hermansky, H., Ellis, D. P. W., and Sharma, S. (2000). Tandem connectionist feature extraction for conventional HMM systems. In *ICASSP*, pages 1635–1638, Istanbul, Turkey.

[66] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

[67] Hinton, G. E., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

[68] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

[69] Hinton, G. E., Vinyals, O., and Dean, J. (2014). Distilling the knowledge in a neural network. In *Deep Learning and Representation Learning Workshop, NIPS*, Montréal, Canada.

[70] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[71] Huang, M., You, Y., Chen, Z., Qian, Y., and Yu, K. (2018). Knowledge distillation for sequence model. In *Interspeech*, pages 3703–3707, Hyderabad, India.

[72] Hyafil, L. and Rivest, R. L. (1976). Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1):15–17.

[73] Jaeger, H. (2010). The "echo state" approach to analysing and training recurrent neural networks. Technical report, Fraunhofer Institute for Autonomous Intelligent Systems.

[74] Jelinek, F. and Mercer, R. (1980). Interpolated estimation of Markov source parameters from sparse data. In *Pattern Recognition in Practice*.

[75] Joy, N. M., Kothinti, S. R., Umesh, S., and Abraham, B. (2017). Generalized distillation framework for speaker normalization. In *Interspeech*, pages 739–743, Stockholm, Sweden.

[76] Juang, B.-H. (1985). Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Technical Journal*, 64(6):1235–1249.

[77] Kaiser, J., Horvat, B., and Kačič, Z. (2002). Overall risk criterion estimation of hidden Markov model parameters. *Speech Communication*, 38(3-4):383–398.

[78] Kanda, N., Fujita, Y., and Nagamatsu, K. (2017). Investigation of lattice-free maximum mutual information-based acoustic models with sequence-level Kullback-Leibler divergence. In *ASRU*, pages 69–76, Okinawa, Japan.

[79] Kanda, N., Fujita, Y., and Nagamatsu, K. (2018). Sequence distillation for purely sequence trained acoustic models. In *ICASSP*, pages 5964–5968, Calgary, Canada.

[80] Kanda, N., Lu, X., and Kawai, H. (2016). Maximum a posteriori based decoding for CTC acoustic models. In *Interspeech*, pages 1868–1872, San Francisco, USA.

[81] Kaplan, R. M. and Kay, M. (1994). Regular models of phonological rule systems. *Computational Linguistics - special issue on computational phonology*, 20(3):331–378.

[82] Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401.

[83] Killer, M., Stüker, S., and Schultz, T. (2003). Grapheme based speech recognition. In *Interspeech*, pages 3141–3144, Geneva, Switzerland.

[84] Kim, D. Y., Chan, H. Y., Evermann, G., Gales, M. J. F., Mrva, D., Sim, K. C., and Woodland, P. C. (2005). Development of the CU-HTK 2004 broadcast news transcription systems. In *ICASSP*, pages 861–864, Philadelphia, USA.

[85] Kim, J., El-Khamy, M., and Lee, J. (2018). BRIDGENETS: student-teacher transfer learning based on recursive neural networks and its application to distant speech recognition. In *ICASSP*, pages 5719–5723, Calgary, Canada.

[86] Kingsbury, B. (2009). Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling. In *ICASSP*, pages 3761–3764, Taipei.

[87] Kirchhoff, K. and Bilmes, J. A. (2000). Combination and joint training of acoustic classifiers for speech recognition. In *Automatic Speech Recognition: challenges for the new millenium*, Paris, France.

[88] Kneser, R. and Ney, H. (1995). Improved back-off for M-gram language modeling. In *ICASSP*, pages 181–184, Detroit, USA.

[89] Lamel, L., Gauvain, J.-L., and Adda, G. (2000). Lightly supervised acoustic model training. In *ASR-2000*, pages 150–154, Paris, France.

[90] Lanchantin, P., Gales, M. J. F., Karanasou, P., Liu, X., Qian, Y., Wang, L., Woodland, P. C., and Zhang, C. (2016). Selection of multi-genre broadcast data for the training of automatic speech recognition systems. In *Interspeech*, pages 3057–3061, San Francisco, USA.

[91] Li, C., Chen, C., Carlson, D., and Carin, L. (2016). Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *AAAI*, pages 1788–1794, Phoenix, USA.

[92] Li, J., Seltzer, M. L., Wang, X., Zhao, R., and Gong, Y. (2017). Large-scale domain adaptation via teacher-student learning. In *Interspeech*, pages 2386–2390, Stockholm, Sweden.

[93] Li, J., Zhao, R., Huang, J.-T., and Gong, Y. (2014). Learning small-size DNN with output-distribution-based criteria. In *Interspeech*, pages 1910–1914, Singapore.

[94] Li, Z. and Eisner, J. (2009). First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Empirical Methods on Natural Language Processing*, pages 40–51, Singapore.

[95] Lippmann, R. P. (1997). Speech recognition by machines and humans. *Speech Communications*, 22(1):1–15.

[96] liu, X., Wang, Y., Chen, X., Gales, M. J. F., and Woodland, P. C. (2014). Efficient lattice rescoring using recurrent neural network language models. In *ICASSP*, pages 4908–4912, Florence, Italy.

[97] MacKay, D. J. C. (1992a). Bayesian interpolation. *Neural Computation*, 4(3):415–447.

[98] MacKay, D. J. C. (1992b). The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736.

[99] Mackay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge University Press.

[100] Mangu, L., Brill, E., and Stolcke, A. (2000). Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400.

[101] Mermelstein, P. (1976). Distance measures for speech recognition, psychological and instrumental. In Chen, C. H., editor, *Pattern Recognition and Artificial Intelligence*, pages 374–388. Academic Press.

[102] Mikolov, T., Karafiát, M., Burget, L., Černocký, J. H., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech*, pages 1045–1048, Makuhari, Japan.

[103] Mikolov, T., Kombrink, S., Deoras, A., Burget, L., and Černocký, J. H. (2011). RNNLM - recurrent neural network language modeling toolkit. In *ASRU*, Big Island, Hawaii, USA.

[104] Mohan, A. and Rose, R. (2015). Multi-lingual speech recognition with low-rank multi-task deep neural networks. In *ICASSP*, pages 4994–4998, Brisbane, Australia.

[105] Morin, F. and Bengio, Y. (2005). Hierarchical probabilistic neural network language model. In *AISTATS*, pages 246–252, Barbados.

[106] Neal, R. M. (1992). Bayesian learning via stochastic dynamics. In *NIPS*, pages 475–482, Denver, USA.

[107] Normandin, Y. (1991). *Hidden Markov models, maximum mutual information estimation, and the speech recognition problem*. PhD thesis, McGill University.

[108] Normandin, Y., Lacouture, R., and Cardin, R. (1994). MMIE training for large vocabulary continuous speech recognition. In *ICSLP*, pages 1367–1370, Yokohama, Japan.

[109] Ozturk, M. C. (2007). *Dynamical computation with echo state networks*. PhD thesis, University of Florida.

[110] Palaz, D., Collobert, R., and Magimai-Doss, M. (2013). Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. In *Interspeech*, pages 1766–1770, Lyon, France.

[111] Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *ICML*, pages 84–90, Atlanta, USA.

[112] Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Interspeech*, pages 3214–3218, Dresden, Germany.

[113] Povey, D. (2003). *Discriminative training for large vocabulary speech recognition*. PhD thesis, University of Cambridge.

[114] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlíček, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G., and Veselý, K. (2011). The Kaldi speech recognition toolkit. In *ASRU*, Hawaii, USA.

[115] Povey, D. and Kingsbury, B. (2007). Evaluation of proposed modifications to MPE for large scale discriminative training. In *ICASSP*, pages 321–324, Honolulu, USA.

[116] Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. (2016). Purely sequence-trained neural networks for ASR based on lattice-free MMI. In *Interspeech*, pages 2751–2755, San Francisco, USA.

[117] Povey, D. and Woodland, P. C. (2002). Minimum phone error and I-smoothing for improved discriminative training. In *ICASSP*, pages 105–108, Orlando, USA.

[118] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[119] Renals, S., Morgan, N., Cohen, M., and Franco, H. (1992). Connectionist probability estimation in the DECIPHER speech recognition system. In *ICASSP*, pages 601–604, San Francisco, USA.

[120] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407.

[121] Robinson, A. J. (1994). An application of recurrent nets to phone probability estimation. *IEEE Transactions on neural networks*, 5(2):298–305.

[122] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. (2015). FITNETS: hints for thin deep nets. In *ICLR*, San Diego, USA.

[123] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536.

[124] Sainath, T. N., Kingsbury, B., Sindhwani, V., Arisoy, E., and Ramabhadran, B. (2013). Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *ICASSP*, pages 6655–6659, Vancouver, Canada.

[125] Sak, H., Senior, A., and Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Interspeech*, pages 338–342, Singapore.

[126] Schlüter, R., Bezrukov, I., Wagner, H., and Ney, H. (2007). Gammatone features and feature combination for large vocabulary speech recognition. In *ICASSP*, pages 649–652, Honolulu, USA.

[127] Schlüter, R., Müller, B., Wessel, F., and Ney, H. (1999). Interdependence of language models and discriminative training. In *ASRU*, pages 119–122, Keystone, USA.

[128] Schlüter, R. and Ney, H. (2001). Using phase spectrum information for improved speech recognition performance. In *ICASSP*, pages 133–136, Salt Lake City, USA.

[129] Sejnowski, T. J. and Rosenberg, C. R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems*, 1(1):145–168.

[130] Seltzer, M. L. and Droppo, J. (2013). Multi-task learning in deep neural networks for improved phoneme recognition. In *ICASSP*, pages 6965–6969, Vancouver, Canada.

[131] Sennrich, R., Haddow, B., and Birch, A. (2016). Edinburgh neural machine translation systems for WMT 16. In *Machine Translation*, pages 371–376, Berlin, Germany.

[132] Shannon, M. (2017). Optimizing expected word error rate via sampling for speech recognition. In *Interspeech*, pages 3537–3541, Stockholm, Sweden.

[133] Shi, Y., Zhang, W.-Q., Cai, M., and Liu, J. (2014). Efficient one-pass decoding with NNLM for speech recognition. *Signal Processing Letters*, 21(4):377–381.

[134] Siohan, O. (2016). Sequence training of multi-task acoustic models using meta-state labels. In *ICASSP*, pages 5425–5429, Shanghai, China.

[135] Siohan, O., Ramabhadran, B., and Kingsbury, B. (2005). Constructing ensembles of ASR systems using randomized decision trees. In *ICASSP*, pages 197–200, Philadelphia, USA.

[136] Siohan, O. and Rybach, D. (2015). Multitask learning and system combination for automatic speech recognition. In *ASRU*, pages 589–595, Scottsdale, USA.

[137] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

[138] Stolcke, A., König, Y., and Weintraub, M. (1997). Explicit word error minimization in N-best list rescoring. In *Eurospeech*, pages 163–166, Rhodes, Greece.

[139] Swietojanski, P., Ghoshal, A., and Renals, S. (2013). Revisiting hybrid and GMM-HMM system combination techniques. In *ICASSP*, pages 6744–6748, Vancouver, Canada.

[140] Takiguchi, T., Bilmes, J., Yoshii, M., and Ariki, Y. (2010). Evaluation of random-projection-based feature combination on speech recognition. In *ICASSP*, pages 2150–2153, Dallas, USA.

[141] Utans, J. (1996). Weight averaging for neural networks and local resampling schemes. In *AAAI Workshop on Integrating Multiple Learned Models for Improving*, pages 133–138, Portland, USA.

[142] Valtchev, V., Odell, J. J., Woodland, P. C., and Young, S. J. (1997). MMIE training of large vocabulary recognition systems. *Speech Communication*, 22(4):303–314.

[143] Veselý, K., Ghoshal, A., Burget, L., and Povey, D. (2013). Sequence-discriminative training of deep neural networks. In *Interspeech*, pages 2345–2349, Lyon, France.

[144] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103, Helsinki, Finland.

[145] Vinyals, O., Ravuri, S. V., and Povey, D. (2012). Revisiting recurrent neural networks for robust ASR. In *ICASSP*, pages 4085–4088, Kyoto, Japan.

[146] Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

[147] Waibel, A., Hanazawa, T., Hinton, G. E., Shikano, K., and Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339.

[148] Wang, H., Ragni, A., Gales, M. J. F., Knill, K. M., Woodland, P. C., and Zhang, C. (2015). Joint decoding of tandem and hybrid systems for improved keyword spotting on low resource languages. In *Interspeech*, pages 3660–3664, Dresden, Germany.

[149] Wang, L., Zhang, C., Woodland, P. C., Gales, M. J. F., Karanasou, P., Lanchantin, P., Liu, X., and Qian, Y. (2016). Improved DNN-based segmentation for multi-genre broadcast audio. In *ICASSP*, pages 5700–5704, Shanghai, China.

[150] Wang, Y., Chen, X., Gales, M. J. F., Ragni, A., and Wong, J. H. M. (2018). Phonetic and graphemic systems for multi-genre broadcast transcription. In *ICASSP*, pages 5899–5903, Calgary, Canada.

[151] Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *ICML*, pages 681–688, Bellevue, USA.

[152] Weng, C., Yu, D., Watanabe, S., and Juang, B.-H. (2014). Recurrent deep neural networks for robust speech recognition. In *ICASSP*, pages 5532–5536, Florence, Italy.

[153] Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356.

[154] Wong, J. H. M. and Gales, M. J. F. (2016). Sequence student-teacher training of deep neural networks. In *Interspeech*, pages 2761–2765, San Francisco, USA.

[155] Wong, J. H. M. and Gales, M. J. F. (2017a). Multi-task ensembles with teacher-student training. In *ASRU*, pages 84–90, Okinawa, Japan.

[156] Wong, J. H. M. and Gales, M. J. F. (2017b). Student-teacher training with diverse decision tree ensembles. In *Interspeech*, pages 117–121, Stockholm, Sweden.

[157] Woodland, P. C. (2002). The development of the HTK broadcast news transcription system: an overview. *Speech Communication*, 37(1-2):47–67.

[158] Woodland, P. C., Gales, M. J. F., Pye, D., and Young, S. J. (1997). The development of the 1996 HTK broadcast news transcription system. In *DARPA Speech Recognition Workshop*, pages 73–78, Chantilly, USA.

[159] Woodland, P. C., Leggetter, C. J., Odell, J. J., Valtchev, V., and Young, S. J. (1995). The 1994 HTK large vocabulary speech recognition system. In *ICASSP*, pages 73–76, Detroit, USA.

[160] Woodland, P. C. and Povey, D. (2002). Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech and Language*, 16(1):25–47.

[161] Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M. L., Stolcke, A., Yu, D., and Zweig, G. (2017). Toward human parity in conversational speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2410–2423.

[162] Xu, H., Chen, G., Povey, D., and Khudanpur, S. (2015). Modeling phonetic context with non-random forests for speech recognition. In *Interspeech*, pages 2117–2121, Dresden, Germany.

[163] Xu, H., Povey, D., Mangu, L., and Zhu, J. (2011). Minimum Bayes risk decoding and system combination based on a recursion for edit distance. *Computer Speech and Language*, 25(4):802–828.

[164] Xue, J., Li, J., and Gong, Y. (2013). Restructuring of deep neural network acoustic models with singular value decomposition. In *Interspeech*, pages 2365–2369, Lyon, France.

[165] Xue, J. and Zhao, Y. (2008). Random forests of phonetic decision trees for acoustic modeling in conversational speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):519–528.

[166] Yang, J., Ragni, A., Gales, M. J. F., and Knill, K. M. (2016). Log-linear system combination using structured support vector machines. In *Interspeech*, pages 1898–1902, San Francisco, USA.

[167] Yildiz, I. B., Jaeger, H., and Kiebel, S. J. (2012). Re-visiting the echo state property. *Neural Networks*, 35:1–9.

[168] Young, S., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J. J., Ollason, D., Povey, D., Ragni, A., Valtchev, V., Woodland, P. C., and Zhang, C. (2015). *The HTK book*.

[169] Young, S. J., Odell, J. J., and Woodland, P. C. (1994). Tree-based state tying for high accuracy acoustic modelling. In *HLT*, pages 307–312, Plainsboro, USA.

[170] Zhan, P. and Waibel, A. (1997). Vocal tract length normalization for large vocabulary continuous speech recognition. Technical report, Carnegie Mellon University.

[171] Zhang, X., Povey, D., and Khudanpur, S. (2015). A diversity-penalizing ensemble training method for deep learning. In *Interspeech*, pages 3590–3594, Dresden, Germany.

[172] Zhao, T., Zhao, Y., and Chen, X. (2014). Building an ensemble of CD-DNN-HMM acoustic model using random forests of phonetic decision trees. In *ISCSLP*, pages 98–102, Singapore.